Theoretical Computer Science 151 (1995) 207–256

Theoretical Computer Science

Full abstraction and recursion

Michael W. Mislove [a,*,1], Frank J. Oles [b]

[a] *Department of Mathematics, Tulane University, New Orleans, LA 70118, USA*
[b] *Mathematical Sciences Department, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA*

Abstract

The question of extending semantic models for a programming language without recursion to a larger language supporting recursion is of fundamental importance if one wants to develop a modular approach to building models. In our previous work [14] we showed how to extend models for a language without recursion to models for a language in which unnested recursion was supported via simultaneous systems of recursive equations. In that setting, we proved that modest assumptions about the models enabled us to lift adequacy and full abstraction for the models of the base language to analogous results for the models of the extended language.

In this paper, we tackle similar problems for languages with explicit recursion operators. Starting with a base language L, we define what it means for an ordered denotational model for L to be *order adequate* and *strongly order fully abstract* with respect to an ordered operational model for L. By freely adding identifiers $x \in X$ and recursion operators μx for each $x \in X$, we obtain an extended language $L\{X\}$, in which we single out the sublanguage L_c of closed terms. We then give general conditions which ensure that an ordered operational model $\mathcal{O} : L \to O$ and a related order adequate and strongly order fully abstract denotational model $\mathcal{D} : L \to D$ can be extended to an ordered operational model $\mathcal{O}_c : L_c \to O$ and a related ordered denotational model $\mathcal{D}_c : L_c \to D$ which also is order adequate and strongly order fully abstract.

1. Introduction

Programming languages are most often presented using a BNF-like set of production rules, i.e., they are most often treated as grammars. From a mathematical viewpoint, this abstract syntax can be regarded as the signature of an initial universal algebra which is the language. This has the added virtue of allowing a well-developed collection of mathematical results to be applied to the language. If the language also is meant to satisfy certain laws – e.g., that some operations are associative and/or commutative – then this amounts to assuming that the algebra is an initial algebra with the given signature in the category of algebras satisfying these laws. Equivalently, we could

* Corresponding author. Email: mwm@math.tulane.edu.
[1] Work partially supported by the Office of Naval Research.

SSDI 0304-3975(95)00052-6

consider the language L to be a quotient of the initial anomic algebra (i.e., an algebra satisfying no equational laws, [18]), modulo the laws we wish to impose.

If L is the language under study, then by an *operational model* for L we simply mean a mapping $\mathcal{O} : L \to O$ to some space O of *observations* which can be made of programs in L. In our development, we are not concerned with where this space O comes from or how the mapping $\mathcal{O}$ is defined. A *denotational model* of L is any algebra D with the same signature as L together with an algebra homomorphism $\mathcal{D} : L \to D$. A *context* is a one-variable polynomial over the algebra L, and the algebra of contexts over L is denoted $L[*]$. If $\mathcal{O}$ is an operational model and $\mathcal{D}$ is a denotational model for L, then the denotational model $\mathcal{D}$ is *adequate and fully abstract* with respect to $\mathcal{O}$ if, for all $p, q \in L$, the following conditions are equivalent:

(i) $\mathcal{D}(p) = \mathcal{D}(q)$.

(ii) $\mathcal{O}(C[p]) = \mathcal{O}(C[q])$ for all contexts $C \in L[*]$.

Viewing languages as algebras naturally leads to the view that the language is built up from basic constituents such as the constants which are incorporated in the language, the operators of the language, the identifiers and the recursive terms which the language supports, etc. So, it is natural to seek a modular approach to building models to support these various constituents. In particular, given a programming language L without identifiers together with an operational model for L and a related denotational model which is adequate and fully abstract with respect to the given operational model, we are interested in establishing conditions that allow us to extend the models for L to models for an extended language L' which includes identifiers and some form of recursion so that the relationship of adequacy and full abstraction is preserved.

If X is an infinite set of identifiers, then one approach is to extend the language L to a language $L[X]$ by adding only identifiers to L, with the understanding that the meaning of an identifier is to be extracted from a system of recursive equations. In [14], it is shown that adequacy and full abstraction for the models of L can be lifted to models for $L[X]$ under fairly mild assumptions, namely, that

(i) the denotational model for L is an algebraic cpo endowed with Scott continuous interpretations of the operations, and

(ii) the operational model for L is a Hausdorff space in which the behavior of a recursive term is the limit of its finite truncations.

All of this worked out fairly nicely as an exercise in universal algebra and category theory, and the range of languages to which the results in [14] apply is reasonably broad (cf., in particular, [2]).

An equally popular approach to adding recursion to a language is via recursion operators. In this approach, one simultaneously adds the identifiers $x \in X$ (thus incorporating the algebra $L[X]$) and unary recursion operators μx for each $x \in X$ to form the algebra $L\{X\}$. The focus then becomes the semantics of the subset L_c of $L\{X\}$ of *closed terms,* i.e., those terms in which all occurrences of identifiers are bound. The reason for this restriction is that the operational meanings of terms that are not closed are less clear. In this paper we explore conditions on the models for L that are sufficient to lift adequacy and full abstraction for them to models for L_c.

A delicate point is whether we should restrict L_c further to a smaller language of what might be called *guarded* closed terms, in which only guarded recursions (cf. [4]) are permitted. We do not impose such a restriction for the following reasons. In the general algebraic setting, there is no clear definition of "guarded".[2] Usually, guardedness is introduced to facilitate the definition of a structural operational semantics by eliminating problematic terms from consideration. But, the main results of this paper assume the operational model is a cpo, and this enables us to assign a reasonable operational meaning to every closed term of the extended language, based on the operational semantics of the base language. This eliminates the need for any further restrictions on L_c.

Since programming languages have algebraic structure, the desire for a compositional semantics leads us to investigate models that preserve that algebraic structure. Moreover, one way to understand recursion is to assume that the programming language has an *undefined program* that can be used to define unwindings of recursive terms. Since it is algebraically simple to add an "undefined program" to any language, if it is not there already, this leads us to assume that there is a constant in the signature for L corresponding to the undefined program. It follows that any such algebra automatically comes with a pre-order that we call the *minimum monotone pre-order*. And, since L is pre-ordered, it is natural to assume that *both* O and D are posets, that $\mathcal{O} : L \to O$ and $\mathcal{D} : L \to D$ are monotone, and that the operations on the algebra D are monotone. This is what we mean by an *ordered operational model* and an *ordered denotational model* for L. It is easy to show that any homomorphism of L into an ordered algebra preserves the minimum monotone pre-order. However, monotonicity of the mapping $\mathcal{O} : L \to O$ is a *requirement* that must be added to the definition of an ordered operational model. This monotonicity is the essential ingredient that distinguishes an ordered operational model from an arbitrary function into an ordered space.

Given an ordered operational model $\mathcal{O}$ and an ordered denotational model $\mathcal{D}$ for L, it is natural to seek some relationship between $\mathcal{D}$ and $\mathcal{O}$ that involves the orders on each of D and O. We therefore formulate a strengthened version of adequacy and full abstraction appropriate to ordered models. We say that an ordered denotational model $\mathcal{D}$ is *order adequate and order fully abstract* with respect to an ordered operational model $\mathcal{O}$ if, for all $p, q \in L$, the following conditions are equivalent:

(i) $\mathcal{D}(p) \leqslant_D \mathcal{D}(q)$.

(ii) $\mathcal{O}(C[p]) \leqslant_O \mathcal{O}(C[q])$, for all contexts $C \in L[*]$.

Thus, our goal is to give conditions that enable us to extend ordered models $\mathcal{O} : L \to O$ and $\mathcal{D} : L \to D$ of L to ordered models $\mathcal{O}_c : L_c \to O$ and $\mathcal{D}_c : L_c \to D$ so that order adequacy and order full abstraction for the models of L imply the same properties for the models of L_c.

[2] One of the referees for this paper pointed out the fairly general definition of guardedness for algebras with a "separating family of congruences" from [19]. However, we do not see how to incorporate this definition into our setting to obtain stronger or more interesting results.

If D is a cpo and the operations on D are Scott continuous, then we can construct $\mathscr{D}_c$ from $\mathscr{D}$. And, if O is a cpo, then we can construct $\mathscr{O}_c$ from $\mathscr{O}$, as follows. For each closed term $q \in L_c$, the undefined program allows us to define a sequence of syntactic unwindings $\pi_0(q), \pi_1(q), \pi_2(q), \ldots$, which are elements of the base language L (cf. Proposition 2.18). These syntactic unwindings allow us to define $\mathscr{O}_c : L_c \to O$ by $\mathscr{O}_c(p) = \bigsqcup\{\mathscr{O}(\pi_n(p)) \mid n \in \mathbb{N}\}$ (cf. Proposition 5.2). These unwindings also are used to state two more key conditions on ordered models that intuitively amount to assuming that the base language L consists only of terminating programs:

(i) For all $p \in L$ and $q \in L_c$, $\mathscr{D}(p) \leqslant_D \bigsqcup\{\mathscr{D}(\pi_n(q)) | n \in \mathbb{N}\}$ implies there exists some $m \in \mathbb{N}$ such that $\mathscr{D}(p) \leqslant_D \mathscr{D}(\pi_m(q))$.

(ii) For all $p \in L$ and $q \in L_c$, $\mathscr{O}(p) \leqslant_O \bigsqcup\{\mathscr{O}(\pi_n(q)) | n \in \mathbb{N}\}$ implies there exists some $m \in \mathbb{N}$ such that $\mathscr{O}(p) \leqslant_O \mathscr{O}(\pi_m(q))$.

Note that condition (i) holds if $\mathscr{D}(L) \subseteq K(D)$, and condition (ii) holds if $\mathscr{O}(L) \subseteq K(O)$, where $K(D)$ and $K(O)$ are the sets of compact elements of the respective cpo's. Since these clearly are finiteness conditions, so we call condition (i) the *denotational finiteness condition* and condition (ii) the *operational finiteness condition.*

Unfortunately, the requirements listed so far are not quite strong enough to yield the results we want. To be sure, the listed conditions are enough to prove that the order adequacy of $\mathscr{D}$ with respect to $\mathscr{O}$ implies the order adequacy of $\mathscr{D}_c$ with respect to $\mathscr{O}_c$. But, even if the finiteness conditions hold, the question of whether $\mathscr{D}_c$ is order fully abstract with respect to $\mathscr{O}_c$when the same relationship holds between $\mathscr{D}$ and $\mathscr{O}$ is a problem. Neither can we prove this result, nor do we have a counterexample. So, in addition to the hypotheses listed above, we need the assumption that $\mathscr{D}$ is a *strongly order fully abstract denotational model* for $\mathscr{O}$. If $\preceq_L$ denotes the minimum monotone pre-order on L, this notion can be stated as,

- $\mathscr{D}$ is *strongly order fully abstract* with respect to $\mathscr{O}$ if, given $p, q \in L$ with $\mathscr{D}(p) \not\leqslant_D \mathscr{D}(q)$, there is a context $C \in L[*]$ such that, whenever $q \preceq_L q'$, then $\mathscr{O}(C[p]) \not\leqslant_O \mathscr{O}(C[q'])$.

Our results show that conditions (i) and (ii) above ensure that if $\mathscr{D}$ is order adequate and strongly order fully abstract with respect to $\mathscr{O}$, then the denotational model $\mathscr{D}_c : L_c \to D$ we construct is order adequate and strongly order fully abstract with respect to the operational model $\mathscr{O}_c : L_c \to O$ that we also construct (cf. Theorems 6.1, 6.3 and 7.6). We do not know of examples that show that the finiteness conditions for the base language are necessary in order to lift the adequacy and full abstraction results to an extended language supporting recursion.

In the process of presenting our results, we include an example simple language without identifiers or recursion. The purpose is to demonstrate how the results can be applied to build an operational model and related strongly order fully abstract denotational model for the closed terms of the simple language extended to include identifiers and recursion operators. Our results also can be applied to show how identifiers and recursion operators can be added to the language studied in [13, 12]. All the proofs apply to denotational models that are algebraic bounded complete sup-semilattices, so the results also apply to the language studied in [16]. Finally, the techniques we develop rely to some extent on ideas put forth in [8] (see, for example, the definition of the mappings $\{\pi_n \mid n \in \mathbb{N}\}$ preceding Proposition 2.15), and in fact they provide an alternative method for obtaining full abstraction results for the language with recursion studied in [8]. This is discussed in more detail in Section 7.

There are some negative results in the literature related to the problem we are studying. The question of a "fair" semantics for countable nondeterminism was studied in [1], where it was shown that a fully abstract model supporting recursion was not attainable with respect to the given fair operational semantics for the language being studied. These results do not contradict ours. Instead, they show that any attempt to validate the assumptions that make our results go through are doomed to fail when considering fair semantics for countable nondeterminism. Issues related to our approach also are addressed in [5].

The organization of the paper is as follows. In the next section we describe the general setting in which we derive our results, and we also present several results about universal algebras and ordered algebras. Then, in Section 3 we derive some results which relate more closely to semantics, and we state precisely the assumptions we require on the operational and denotational models of the base language L with which we begin. We also present the definitions of order adequacy and (strong) order full abstraction. These definitions generalize the notions of adequacy and full abstraction which generally are used, and they seem most appropriate for the setting in which we find ourselves. In Section 4 we derive the denotational model for the language L_c, and in Section 5 we derive the operational model for L_c. Then, in Section 6 we give conditions ensuring the denotational model for L_c derived in Section 4 is order adequate and strongly order fully abstract for the operational model given in Section 5. Finally, in the Section 7, we extend the results of Section 6 to languages with recursion that satisfy a set of equations, thereby greatly expanding the applicability of the work here.

Finally, we want to thank the referees for very careful reading of the first version of this paper, and for their suggestions which have improved the presentation considerably.

2. Languages and ordered algebras

We already have pointed out that there are certain advantages to presenting programming languages as algebras, both for reasons of elegance and to take advantage of the well-developed concepts of universal algebra in technical arguments. For simplicity, we assume we are given a basic language L which is presented as an Ω-algebra for a one-sorted (as opposed to a many-sorted) signature Ω. There is no need to make any assumption about exactly what operators are in Ω, except that it contains at least one constant b, to provide a way of referring to the *undefined program* (denoted by $b_L \in L$). In particular, we do not assume that L is an initial or free anomic Ω-algebra, so L could be an Ω-algebra satisfying some set of equations. In fact, the example language we introduce below is of this type.

Example 2.1. We now begin the presentation of a simple language that will serve to illustrate the application of our results. The language S we have in mind could be described as having the following set of informal BNF-like production rules:

$$p ::= b_S \mid \delta_S \mid \varepsilon_S \mid a \mid p\,;q \mid p+q.$$

The term b_S denotes the *undefined program* in our language. The constant δ_S denotes a program which is deadlocked and of which no observable behavior is possible; this program sometimes is referred to as immediate abnormal termination. On the other hand, the constant ε_S denotes immediate normal termination. The constants a, where a ranges over the set A of atomic actions, denote processes that first execute the action a, and then normally terminate. The binary operator ; denotes sequential composition, and $+$ denotes nondeterministic choice. This language incorporates constructs used in process algebra and in concurrency, except it does not support any form of communication or synchronization.

A more formal presentation of our language S begins with the introduction of the signature Σ :

$$\Sigma_0 = \{b, \delta, \varepsilon\} \cup \{a \mid a \in A\}; \qquad \Sigma_2 = \{;, +\}; \qquad \Sigma_n = \emptyset \text{ if } n \neq 0, 2.$$

Here is the set of equations E for Σ we also want to hold:

(i) ; is associative.
(ii) ε is a two-sided identity for ;.
(iii) $+$ is associative, commutative and idempotent.
(iv) δ is a two-sided identity for $+$.
(v) δ is a left zero for ;.
(vi) $(x+y);z = (x;z)+(y;z)$, i.e., *right distributivity* of ; over $+$.

Then we can view S as the initial (Σ, E)-algebra; i.e., S is the initial Σ-algebra satisfying the equations (i)–(vi).

We are concerned with models for the language L which are ordered spaces, and so we establish some results about ordered Ω-algebras.

Definition 2.2. An *ordered Ω-algebra* is an Ω-algebra A which also is a partially ordered space with least element $\perp_A$ such that

(i) $b_A = \perp_A$, and
(ii) for each $\omega \in \Omega$, the mapping $\omega : A^n \to A$ is monotone, where n is the arity of ω and we equip A^n with the product order.

It is customary to consider as a denotational model for a language with signature Ω a *continuous Ω-algebra,* i.e, a cpo that is an Ω-algebra relative to which the operations of Ω are Scott continuous. Clearly any such model is an ordered Ω-algebra.

We call a pre-order $\preceq$ on an Ω-algebra A *monotone* if, for all $a \in A$, $b_A \preceq a$, and the interpretations of all the operators from Ω are monotone with respect to $\preceq$. The family of all monotone pre-orders is not empty, since the universal relation is one such pre-order. And, the intersection of this family is another such pre-order, $\preceq_A$, and clearly it is minimal. The pre-order $\preceq_A$ is known as the *minimum monotone pre-order* on A. This concept was first employed in work relating non-well-founded sets and domain theory found in [11].

Proposition 2.3. *For all elements a and a' of an Ω-algebra A, the following are equivalent*:

(i) $a \preceq_A a'$.

(ii) *$\phi a \leqslant_B \phi a'$ for any Ω-algebra homomorphism $\phi : A \to B$ to an ordered Ω-algebra B.*

Proof. If $\phi : A \to B$ is an Ω-algebra map into an ordered algebra, then the partial order on B can be pulled back to a pre-order on A defined by $a \preceq_\phi a'$ if and only if $\phi a \leqslant_B \phi a'$. Note, in particular, that $\phi b_A \leqslant_B \phi a$ for each $a \in A$ since $\phi b_A = b_B = \perp_B$, the last equality following because B is an ordered Ω-algebra. Hence $b_A \preceq_\phi a$ for each $a \in A$, and it follows that $\preceq_\phi$ is a monotone pre-order on A. Because of the minimality of $\preceq_A$, $\preceq_A \subseteq \preceq_\phi$; i.e., $a \preceq_A a'$ implies $\phi a \leqslant_B \phi a'$.

For the converse, note that the minimum monotone pre-order on A gives rise to a surjective homomorphism onto the algebra $A/\equiv$ of equivalence classes induced by $\preceq_A$, where $x \equiv x'$ if and only if $x \preceq_A x'$ and $x' \preceq_A x$. Moreover, the image of $\preceq_A$ under the projection map is a monotone partial order on $A/\equiv$, so that $A/\equiv$ is an ordered algebra. This monotone partial order, when pulled back to A is the minimum monotone pre-order $\preceq_A$. Hence, for $a, a' \in A$, the second condition applied to the projection onto $A/\equiv$ implies that $a \preceq_A a'$. $\square$

Corollary 2.4. *Let $\phi : A \to B$ be an Ω-algebra homomorphism. If $a \preceq_A a'$, then $\phi a \preceq_B \phi a'$.*

Proof. Let $\psi : B \to C$ be an arbitrary homomorphism into an ordered Ω-algebra C, so that $\psi \circ \phi : A \to C$ is a homomorphism. If $a \preceq_A a'$, then $\psi(\phi a) \leqslant_C \psi(\phi a')$. Since ψ was arbitrary, $\phi a \preceq_B \phi a'$. $\square$

Let $\mathscr{C}_\Omega$ be the category of Ω-algebras and Ω-algebra homomorphisms, and let $\mathscr{C}_{(\Omega,\leqslant)}$ be the category of ordered Ω-algebras and monotone Ω-algebra maps. The following corollary is too easy to state to omit.

Corollary 2.5. *$\mathscr{C}_{(\Omega,\leqslant)}$ is a reflective subcategory of $\mathscr{C}_\Omega$, i.e., there is a left adjoint to the inclusion functor from $\mathscr{C}_{(\Omega,\leqslant)}$ to $\mathscr{C}_\Omega$.*

Proof. The unit of the adjunction is the projection from an Ω-algebra A onto the algebra $A/\equiv$ of equivalence classes induced by the minimum monotone pre-order as in the proof of Proposition 2.3. $\square$

We next discuss polynomial algebras in general and algebras of contexts in particular. Let S be a set and let A be an Ω-algebra. The identity homomorphism on A is denoted

$1_A : A \to A$. Let $T_\Omega[S]$ be a free Ω-algebra generated by S. For notational simplicity, we assume that the function $S \to T_\Omega[S]$ that is the unit of the adjunction is actually an inclusion map $S \hookrightarrow T_\Omega[S]$. Just as in the construction of rings of polynomials, S can be added freely to any Ω-algebra A by letting the polynomial algebra $A[S]$ be a coproduct of A and $T_\Omega[S]$. (A proof of the existence of coproducts of Ω-algebras and some applications of coproducts to the theory of programming languages can be found in [20].) Without loss of generality, we assume A is a subalgebra of $A[S]$ and the coproduct injection $A \to A[S]$ is the inclusion $A \hookrightarrow A[S]$. As long as $A \cap S = \emptyset$, which we always tacitly assume whenever it is convenient, we can arrange matters so that the restriction of the coproduct injection $T_\Omega[S] \to A[S]$ to S is the inclusion map $S \hookrightarrow A[S]$. It is easy to see that the polynomial algebra $A[S]$ satisfies the following universal mapping property: for each Ω-algebra B, each homomorphism $\alpha : A \to B$, and each function $f : S \to B$, there exists a unique homomorphism $\alpha[f] : A[S] \to B$ such that the diagram

$$\begin{tikzcd} A \arrow[r, "incl"] \arrow[dr, "\alpha"'] & A[S] \arrow[d, "\alpha[f]"] & S \arrow[l, "incl"'] \arrow[dl, "f"] \\ & B & \end{tikzcd}$$

commutes.

To facilitate inductive arguments, it is useful to define on each polynomial algebra a rank homomorphism $\rho : A[S] \to \mathbb{N}$.

Definition 2.6. Let A be an Ω-algebra and let S be a set. We define the *rank function* $\rho : A[S] \to \mathbb{N}$ as follows: Endow $\mathbb{N}$ with an Ω-structure such that, for an n-ary operator ω,

$$\omega_{\mathbb{N}}\langle m_1, \ldots, m_n\rangle = 3^{\sup\{m_1,\ldots,m_n\}} - 1.$$

Then $\{0\}$ is an Ω-subalgebra of $\mathbb{N}$, and $\rho : A[S] \to \mathbb{N}$ can be defined to be the unique Ω-algebra homomorphism such that

$$\begin{tikzcd} A \arrow[r, "incl"] \arrow[dr, "\lambda a.0"'] & A[S] \arrow[d, "\rho"] & S \arrow[l, "incl"'] \arrow[dl, "\lambda s.1"] \\ & \mathbb{N} & \end{tikzcd}$$

is a commutative diagram.

This rank function $\rho : A[S] \to \mathbb{N}$ can be used to prove (1) implies (2) of the following result, the converse direction of which is clear.

Proposition 2.7. *Let $\alpha : A \to B$ be an Ω-algebra homomorphism into an ordered Ω-algebra B, and let $f, f' : S \to B$ be functions. The following are equivalent*:

(i) $f \leqslant_{(S\to B)} f'$,

(ii) $\alpha[f] \leqslant_{(A[S]\to B)} \alpha[f']$,

where we endow $(S \to B)$ and $(A[S] \to B)$ with the pointwise order.

Proof. (1) implies (2): Let $f, f' : S \to B$ satisfy $f \leqslant_{(S\to B)} f'$, and let $x \in A[S]$. We proceed by induction on $\rho(x)$. If $\rho(x) = 0$, then $x \in A$, and so $\alpha[f]x = \alpha x = \alpha[f']x$.

Suppose that the result holds for all terms $x \in A[S]$ with $\rho(x) \leqslant n \in \mathbb{N}$, and let $x \in A[S]$ satisfy $\rho(x) = n + 1$. Then $x = \omega_m\langle x_1, \dots, x_m\rangle$, and so

$$\begin{aligned}
\alpha[f]x &= \alpha[f]\,\omega_m\langle x_1,\dots,x_m\rangle \\
&= \omega_m\langle \alpha[f]x_1,\dots,\alpha[f]x_m\rangle \\
&\leqslant_B \omega_m\langle \alpha[f']x_1,\dots,\alpha[f']x_m\rangle \\
&= \alpha[f']\,\omega_m\langle x_1,\dots,x_m\rangle = \alpha[f']x,
\end{aligned}$$

the inequality following from the inductive hypothesis. $\square$

Corollary 2.8. *Let $\alpha : A \to B$ be an Ω-algebra homomorphism into an Ω-algebra B, and let $f, f' : S \to B$ be functions such that $fs \preceq_B f's$ for each $s \in S$. Then $\alpha[f]\,p \preceq_B \alpha[f']\,p$ for all $p \in A[S]$.*

Proof. This can be proved by applying the proposition to the composition $\phi \circ \alpha : A \to B'$, where $\phi : B \to B'$ ranges over all homomorphisms into ordered Ω-algebras.

The notion of full abstraction requires the concept of a *context*, i.e., "a program with a hole in it". To be precise, an Ω-algebra of contexts over A is nothing more than the algebra of polynomials $A[*]$ ($= A[\{*\}]$) in one indeterminate $*$ over A.

If $f : A \to B$ is an Ω-algebra homomorphism that is clear from context – typically f is an inclusion into a larger algebra – we can define the substitution of an element $b \in B$ for $*$ in a context $C \in A[*]$ to obtain $C[b] \in B$. To be precise, let $C[b] = f[* \mapsto b]C$, where $f[* \mapsto b]$ is the unique Ω-algebra homomorphism such that

$$\begin{tikzcd}
A \arrow[r, "incl"] \arrow[dr, "f"'] & A[*] \arrow[d, "f[* \mapsto b]"'] & \{*\} \arrow[l, "incl"'] \arrow[dl, "* \mapsto b"] \\
& B &
\end{tikzcd}$$

commutes.

In particular, substitution of $a \in A$ for $*$ in a context $C \in A[*]$ is written $C[a]$ and is defined to be $C[a] = \mathbf{1}_A[* \mapsto a]C$.

However, it is important to note that this definition of $C[a]$ is only appropriate to *algebras whose signatures do not contain identifiers and corresponding binding operators.* The proper definition of substitution into a context containing binding operators will be addressed in Section 4.

That "contexts act monotonically" on an ordered Ω-algebra B is implied by the next proposition.

Proposition 2.9. *Let $f : A \to B$ be an Ω-algebra homomorphism into an ordered Ω-algebra B. For all $b, b' \in B$, the following are equivalent:*

(i) $b \leqslant_B b'$.

(ii) *For all $C \in A[*]$, $f[* \mapsto b]C \leqslant_B f[* \mapsto b']C$.*

Proof. This is just a special case of Proposition 2.7. $\square$

Next we establish in Proposition 2.11 a natural isomorphism that describes the extent to which "contexts commute with homomorphisms". To derive this result, we first recall the notion of the *comma category $\langle A \downarrow \mathscr{C}_\Omega \rangle$ under the algebra A.*

Definition 2.10. If A is an Ω-algebra and $\mathscr{C}_\Omega$ denotes the category of Ω-algebras and Ω-algebra maps, then by $\langle A \downarrow \mathscr{C}_\Omega \rangle$ we mean that category whose objects are pairs $\langle f, B \rangle$, where B is an object of $\mathscr{C}_\Omega$ and $f : A \to B$ is an Ω-algebra map. Given two such objects, $\langle f, B \rangle$ and $\langle f', B' \rangle$, then a $\langle A \downarrow \mathscr{C}_\Omega \rangle$-morphism $g : \langle f, B \rangle \to \langle f', B' \rangle$ is an Ω-algebra homomorphism $g : B \to B'$ making the following diagram commute:

$$\begin{tikzcd} & A \arrow[dl, "f"'] \arrow[dr, "f'"] & \\ B \arrow[rr, "g"] & & B' \end{tikzcd}$$

We can define the "projection on the codomain" $J : \langle A \downarrow \mathscr{C}_\Omega \rangle \to \mathscr{C}_\Omega$ by $J\langle f, B \rangle = B$ and $Jg = g : B \to B'$ if $g : \langle f, B \rangle \to \langle f', B' \rangle$. This functor is related to the functor $F : \langle A \downarrow \mathscr{C}_\Omega \rangle \to \mathscr{C}_\Omega$ given on objects by

$$F\langle f, B \rangle \;=\; \{\phi : A[*] \to B \mid \phi \text{ is an } \Omega\text{-algebra map and } \phi|_A = f\}.$$

The Ω-algebra structure of $F\langle f, B \rangle$ is given by

$$\omega_{F\langle f,B \rangle}\langle \phi_1, \ldots, \phi_m \rangle = \hat{f} = f[* \mapsto \omega_B\langle \phi_1 *, \ldots, \phi_m * \rangle]$$

which fills in the following diagram:

$$\begin{tikzcd}
A \arrow[r, "incl"] \arrow[dr, "f"'] & A[*] \arrow[d, "\widehat{f}"] & \{*\} \arrow[l, "incl"'] \arrow[dl, "* \mapsto \omega_B \langle \phi_1 *, \ldots, \phi_m * \rangle"] \\
& B &
\end{tikzcd}$$

If $g : \langle f,B \rangle \to \langle f',B' \rangle$ is in $\langle A \downarrow \mathscr{C}_\Omega \rangle$, then we define $Fg : F\langle f,B \rangle \to F\langle f',B' \rangle$ by $Fg\,\phi = g \circ \phi$. We note that, since $\phi|_A = f$ and $g \circ f = f'$, the restriction of $g \circ \phi$ to A is f', so Fg is well-defined.

Proposition 2.11. *There is a natural isomorphism* $\eta : J \to F$ *where*

$$\eta_{\langle f,B \rangle} : B \to F\langle f,B \rangle$$

is defined by

$$\eta_{\langle f,B \rangle} b = f[* \mapsto b].$$

Moreover, $\eta^{-1} : F \to J$ *satisfies* $\eta^{-1}_{F\langle f,B \rangle} : F\langle f,B \rangle \to B$ *by* $\eta^{-1}_{F\langle f,B \rangle}\, \phi = \phi\, *$.

Proof. The proof requires showing the following diagram commutes:

$$\begin{array}{ccc}
J\langle f,B \rangle = B & \xrightarrow{\eta_{\langle f,B \rangle}} & F\langle f,B \rangle \\
{\scriptstyle Jg=g}\downarrow & & \downarrow{\scriptstyle \phi \mapsto g \circ \phi} \\
J\langle f',B' \rangle = B' & \xrightarrow[\eta_{\langle f,B \rangle}]{} & F\langle f',B' \rangle
\end{array}$$

This requires us to show that

$$g \circ (\eta_{\langle f,B \rangle} b) = \eta_{\langle f',B' \rangle}(g\,b)$$

for each $b \in B$. Now, the left-hand side of the required equality evaluates as

$$g \circ (\eta_{\langle f,B \rangle} b) = g \circ (f[* \mapsto b]),$$

while the right-hand side of the equality evaluates to

$$\eta_{\langle f',B' \rangle}\, gb = f'[* \mapsto gb].$$

Since $g : \langle f,B \rangle \to \langle f',B' \rangle$ in $\langle A \downarrow \mathscr{C}_\Omega \rangle$, it follows that $f' = g \circ f$. Hence, the required equality reduces to showing that

$$g \circ (f[* \mapsto b]) = (g \circ f)[* \mapsto gb], \tag{1}$$

and this equality follows from inspection of the following diagram:

$$\begin{tikzcd}
A \arrow[r, "incl"] \arrow[dr, "f"] \arrow[ddr, "g \circ f"'] & A[*] \arrow[d, "\hat{f}"] & \{*\} \arrow[l, "incl"'] \arrow[dl, "*_b"'] \arrow[ddl, "*_{gb}"] \\
 & B \arrow[d, "g"] & \\
 & B' &
\end{tikzcd}$$

where $\hat{f} = f[* \mapsto b]$, $*_b$ sends $*$ to the element $b \in B$, and $*_{gb}$ sends it to $gb \in B'$. □

The computational content of the preceding proposition (i.e., the formulation that will be directly used in proofs) is contained in the following corollary, which follows from Eq. (1) in the last part of the proof of the proposition.

Corollary 2.12. *Let $f : A \to B$ and $g : B \to B'$ be Ω-algebra homomorphisms, let $b \in B$, and let $C \in A[*]$. Then*

$$g(f[* \mapsto b]\,C) = (g \circ f)[* \mapsto g\,b]\,C.$$

Remark. As long as $B = B'$ and $f = g \circ f$, we can write the above equation as

$$g(C[b]) = C[g\,b].$$

However, if $f \neq g \circ f$, this equation can be misleading, even if $B = B'$.

We now turn to the analysis of extensions of the language L. Let X be an infinite set of identifiers such that $L \cap X = \emptyset$. The polynomial algebra $L[X]$ is what we get when we extend L by adding the identifiers in X. Given a *syntactic environment* $e : X \to L$, the universal property of $L[X]$ means there is a unique homomorphism $\mathbf{1}_L[e] : L[X] \to L$ such that the diagram

$$\begin{tikzcd}
L \arrow[r, "\iota"] \arrow[dr, "\mathbf{1}_L"'] & L[X] \arrow[d, "\mathbf{1}_L[e]"'] & X \arrow[l, "incl"'] \arrow[dl, "e"] \\
 & L &
\end{tikzcd}$$

commutes, where $\iota : L \to L[X]$ is the inclusion homomorphism. Thus, $\mathbf{1}_L[e]$ evaluates a polynomial in $L[X]$ with the values specified by $e : X \to L$.

Proposition 2.13. *The minimum monotone pre-order on L equals the restriction to L of the minimum monotone pre-order on $L[X]$.*

Proof. Since $L[X]$ is an Ω-algebra, it has a minimal pre-order, $\preceq_{L[X]}$, and since $\iota : L \to L[X]$ is an Ω-algebra homomorphism, it follows that the minimum monotone pre-order on L is coarser than the restriction of the minimum monotone pre-order on $L[X]$ to L. Conversely, Proposition 2.3 implies that the minimum monotone pre-order on L (resp. $L[X]$) is the intersection of the pre-orders induced on L (resp. $L[X]$) from Ω-algebra homomorphisms from L (resp. $L[X]$) into ordered Ω-algebras. Now, given any environment $e : X \to L$ and an Ω-algebra homomorphism $\phi : L \to A$ to an ordered Ω-algebra, we can define the Ω-algebra homomorphism $\phi \circ \mathbf{1}[e] : L[X] \to A$, and the pre-order that this homomorphism induces on L is the restriction to L of the pre-order $\phi \circ \mathbf{1}[e]$ induces on $L[X]$. Hence the restriction of the minimal pre-order on $L[X]$, which is the intersection of the pre-orders induced from Ω-algebra homomorphisms from on $L[X]$ to ordered Ω-algebras, is equal to the minimum monotone pre-order on L, which is the intersection of the pre-orders induced on L by Ω-algebra homomorphisms from L to ordered Ω-algebras. □

Next, we note that substitution of a term $p \in L[X]$ for an identifier $x \in X$ can be defined to be the unique Ω-algebra homomorphism $[p/x] : L[X] \to L[X]$ such that

$$\begin{tikzcd} L \arrow[r, "\iota"] \arrow[dr, "\iota"'] & L[X] \arrow[d, "{[p/x]}"] & X \arrow[l, "incl"'] \arrow[dl, "h"] \\ & L[X] & \end{tikzcd}$$

commutes, where the function $h : X \to L[X]$ is defined by

$$h(y) = \begin{cases} p & \text{if } y = x, \\ y & \text{otherwise.} \end{cases}$$

Of course, we wish to extend L not only by adding identifiers, but also by adding unary binding operators to support recursion. This means we need to extend the signature Ω to $\Omega\{X\}$, defined as follows:

- $\Omega\{X\}_0 = \Omega_0 \cup X$, the constants of Ω together with the identifiers X,
- $\Omega\{X\}_1 = \Omega_1 \cup \{\mu x.- \mid x \in X\}$, the unary operators of Ω together with the recursion operators, and
- $\Omega\{X\}_n = \Omega_n$, for $n \geqslant 2$.

It is well-known that if the signature Ω' is an extension of Ω, then the forgetful functor from the category of Ω'-algebras to the category of Ω-algebras has a left adjoint. Thus, the extended language $L\{X\}$ supporting both identifiers and recursion is characterized by the following universal mapping property: if $f : L \to A$ is any

Ω-algebra homomorphism of L into an $\Omega\{X\}$-algebra A, then there is a unique $\Omega\{X\}$-algebra homomorphism $f_{\Omega\{X\}} : L\{X\} \to A$ so that the following diagram commutes:

$$\begin{tikzcd} L \arrow[r, "\iota"] \arrow[dr, "f"'] & L\{X\} \arrow[d, "f_{\Omega\{X\}}"] \\ & A \end{tikzcd}$$

The structure of $L\{X\}$ simply as an Ω-algebra also is important. In fact, $L\{X\}$ contains $L[X]$ as an Ω-subalgebra, and $L\{X\}$ is a polynomial Ω-algebra over L:$L\{X\}$ satisfies a universal property that is captured by the following diagram, where A is an Ω-algebra, $f : L \to A$ is an Ω-homomorphism, and $g : X \cup \{\mu x.p \mid x \in X,\ p \in L\{X\}\} \to A$ is any function:

$$\begin{tikzcd} L \arrow[r, "\iota"] \arrow[dr, "f"'] & L\{X\} \arrow[d, "f[g]"] & X \cup \{\mu x.p \mid x \in X, p \in L\{X\}\} \arrow[l, "incl"'] \arrow[dl, "g"] \\ & A & \end{tikzcd}$$

That is, $f[g]$ is the unique Ω-algebra homomorphism making the diagram commute.

In order to prove Proposition 2.20 below, we develop a more complete understanding of $\preceq_{L\{X\}}$. When we write $p \preceq_{L\{X\}} q$ for $p, q \in L\{X\}$, we are referring to the minimum monotone pre-order $\preceq_{L\{X\}}$ that is obtained from the structure of $L\{X\}$ as an $\Omega\{X\}$-algebra, and not as an Ω-algebra. The minimum monotone pre-orders arising from the two algebraic structures – Ω and $\Omega\{X\}$ – are not the same. Since L and $L[X]$ have not been endowed with any algebraic structure except as an Ω-algebra, the notations $p \preceq_L q$ and $p \preceq_{L[X]} q$ are unambiguous.

Proposition 2.14. *If $p, q \in L\{X\}$ satisfy $p \preceq_{L\{X\}} q$, then one of the following possibilities holds:*

(i) *Both p and q are in L and $p \preceq_L q$.*

(ii) $p = q$.

(iii) $p = b_L$

(iv) $p = \omega_{L\{X\}}\langle p_1, \ldots, p_n\rangle$ *and* $q = \omega_{L\{X\}}\langle q_1, \ldots, q_n\rangle$, *where* $\omega \in \Omega_n$ *and, for each* $i \in \{1, \ldots, n\}$, $p_i \preceq_{L\{X\}} q_i$.

(v) $p = \mu x_{L\{X\}}.p'$ *and* $q = \mu x_{L\{X\}}.q'$, *where* $x \in X$ *and* $p' \preceq_{L\{X\}} q'$.

Proof. A monotone pre-order on $L\{X\}$ can be thought of as an $\Omega\{X\}$-subalgebra of the product algebra $L\{X\} \times L\{X\}$ that is also a pre-order. It is not hard to see that the $\Omega\{X\}$-subalgebra A of $L\{X\} \times L\{X\}$ generated by the pairs of the form $\langle p, p\rangle$ and the pairs of the form $\langle b_L, p\rangle$, is a pre-order on $L\{X\}$. Since the generators of A are necessarily members of every monotone pre-order on $L\{X\}$, we conclude that A is the minimum monotone pre-order on $L\{X\}$. Thus, this proposition simply summarizes the structure of A as an $\Omega\{X\}$-algebra. □

When we extend L to the language $L\{X\}$ that supports recursion, we want the operational semantics of a recursive program to be obtained by unwinding the recursion in a way that is consistent with the operational semantics of L. We use an idea from [8] as the key to defining the operational meanings of recursive terms. This idea is for each term p with a recursion operator to define a sequence of terms $\pi_n(p)$ without recursion operators whose meanings can be given, and then to define the meaning of the recursive term p to be the supremum of the (increasing) sequence of meanings of the terms $\pi_n(p)$ in the model.[3] Our definition of the π_n's differs somewhat from that of [8], since we need these mappings to be Ω-algebra homomorphisms (e.g., to prove Proposition 2.15). But the underlying motivation remains the same: the $\pi_n(p)$ should correspond to successive unwindings of the recursive term p, with the remaining recursions replaced by the undefined program in each unwinding.

Taking the case that $A = L[X]$ in the last diagram, we inductively define a sequence of Ω-algebra homomorphisms $\pi_n : L\{X\} \to L[X]$ as follows:

- $\pi_0 : L\{X\} \to L[X]$ is the homomorphism determined by
  (i) The inclusion $f = \iota : L \hookrightarrow L[X]$,
  (ii) The inclusion map $x \mapsto x : X \hookrightarrow L[X]$, and
  (iii) The constant map $\mu x.p \mapsto b_L$ for each $x \in X$ and each $p \in L\{X\}$.
- $\pi_{n+1} : L\{X\} \to L[X]$ is the homomorphism determined by
  (i) The inclusion $f = \iota : L \hookrightarrow L[X]$,
  (ii) The inclusion map $x \mapsto x : X \hookrightarrow L[X]$, and
  (iii) The mapping $\mu x.p \mapsto [\pi_n(\mu x.p)/x](\pi_n(p))$ for each $x \in X$ and each $p \in L\{X\}$. Note that the well-definedness of $\pi_{n+1}(p)$ depends on the fact that $\pi_n(\mu x.p)$ actually is in $L[X]$, which ensures that the substitution $[\pi_n(\mu x.p)/x]$ is a self-map of $L[X]$.

A simple application of Corollary 2.12 allows us to conclude that "contexts $C \in L[*]$ commute with the π_n's".

Proposition 2.15. *If $n \in \mathbb{N}$, $C \in L[*]$ and $p \in L\{X\}$, then $\pi_n(C[p]) = C[\pi_n(p)]$.*

Proposition 2.16. *If $p \in L\{X\}$, then $\pi_n(p) \preceq_{L[X]} \pi_{n+1}(p)$ for each $n \in \mathbb{N}$.*

Proof. We proceed by induction on $n \in \mathbb{N}$. If $n = 0$, then, for all $q \in X \cup \{\mu x.p \mid p \in L\{X\}\}$, $\pi_0(q) \preceq_{L[X]} \pi_1(q)$, so the result follows from Corollary 2.8, where

[3] A similar, but more general construction also has been used by Hyland in [9] in the context of the untyped lambda calculus.

$\alpha : L \to L[X]$ is the inclusion and $f, f' : X \cup \{\mu x.p \mid p \in L\{X\}\} \to L[X]$ are given by $fx = f'x = x$, $f(\mu x.p) = b_L$, and $f'(\mu x.p) = [b_L/x](\pi_0(p))$.

Suppose that the result holds for some $n \in \mathbb{N}$, so that $\pi_n(p) \preceq_{L[X]} \pi_{n+1}(p)$ for all $p \in L\{X\}$. Then

$$\begin{aligned}\pi_{n+1}(\mu x.p) &= [\pi_n(\mu x.p)/x](\pi_n(p)) \\ &\preceq_{L[X]} [\pi_{n+1}(\mu x.p)/x](\pi_n(p)) \\ &\preceq_{L[X]} [\pi_{n+1}(\mu x.p)/x](\pi_{n+1}(p)) \\ &= \pi_{n+2}(p),\end{aligned}$$

where

(i) the first $\preceq_{L[X]}$-inequality follows from Corollary 2.8 with $\alpha : L \to L[X]$ the inclusion and f (resp. f') the function sending the identifier x to $\pi_n(\mu x.p)$ (resp. $\pi_{n+1}(\mu x.p)$) and leaving all other identifiers unchanged (as in the proof of the first inductive step) and the inductive hypothesis that $\pi_n(p) \preceq_{L[X]} \pi_{n+1}(p)$ for all $p \in L\{X\}$, and

(ii) the second one follows from the fact that homomorphisms preserve the minimum monotone pre-order, and from the inductive hypothesis. $\square$

As we commented in the introduction, it already has been noted by others that assigning operational meanings to terms in $L\{X\}$ with free variables can be problematical. Thus, we pay special attention to the Ω-subalgebra L_c of closed terms of $L\{X\}$. Again, we rely on algebra to make this notion precise.

Following [15, 17], there is a $\Omega\{X\}$-algebra homomorphism that tells us the set of identifiers that occur freely in a term of $L\{X\}$. To define it, we make the family $\mathscr{P}_{\text{fin}}(X)$ of finite subsets of X into an $\Omega\{X\}$-algebra as follows:

- $\omega_{\mathscr{P}_{\text{fin}}(X)}\langle S_1, \ldots, S_n\rangle = S_1 \cup \cdots \cup S_n$ for each $\omega \in \Omega_n$,
- $x_{\mathscr{P}_{\text{fin}}(X)} = \{x\}$ for each $x \in X$, and
- $\mu x_{\mathscr{P}_{\text{fin}}(X)}.S = S \setminus \{x\}$ for each $x \in X$.

Then we define the function Free : $L\{X\} \to \mathscr{P}_{\text{fin}}(X)$ to be the unique $\Omega\{X\}$-algebra homomorphism which extends the constant Ω-algebra homomorphism $l \mapsto \emptyset : L \to \mathscr{P}_{\text{fin}}(X)$. Note that each finite subset S of X is an Ω-subalgebra of $\mathscr{P}_{\text{fin}}(X)$, so the following definition makes sense.

Definition 2.17. For a finite subset $S \subseteq X$, we define the Ω-subalgebra L_S of $L\{X\}$ to be the family $L_S = \text{Free}^{-1}(S)$. We define the algebra L_c of *closed terms* of $L\{X\}$ to be the Ω-subalgebra $L_c = L_\emptyset$.

Earlier we defined the rank function $\rho : A[S] \to \mathbb{N}$. Before proving the next result, we extend this function to $L\{X\}$, and hence to L_c as well. We first make $\mathbb{N}$ into an $\Omega\{X\}$-algebra by defining:

- $\omega_{\mathbb{N}}\langle m_1, \ldots, m_n\rangle = 3^{\sup\{m_1,\ldots,m_n\}} - 1$ for each $\omega \in \Omega$,

- $x_{\mathbb{N}} = 1$ for each $x \in X$, and
- $\mu x_{\mathbb{N}}.m = m + 2$ for each $x \in X$.

Then the function rank : $L\{X\} \to \mathbb{N}$ is the unique $\Omega\{X\}$-algebra homomorphism which extends the Ω-algebra map sending L to the Ω-subalgebra $\{0\}$ of $\mathbb{N}$. Note that rank is well-defined, since $l = \omega_L\langle l_1, \ldots, l_n\rangle \in L$ implies $\mathrm{rank}(l) = \omega_{\mathbb{N}}\langle \mathrm{rank}(l_1), \ldots, \mathrm{rank}(l_n)\rangle = 3^0 - 1 = 0$. Also note that $\mathrm{rank}(p) = 0$ if and only if $p \in L$, and $\mathrm{rank}(p) = 1$ if and only if $p \in X$.

Proposition 2.18. *For each $p \in L_c$ and each $n \in \mathbb{N}$, the term $\pi_n(p) \in L$.*

Proof. We proceed by induction on $\mathrm{rank}(p)$. If $\mathrm{rank}(p) = 0$, then $p \in L$, and $\pi_n(p) = p \in L$ for each $n \in \mathbb{N}$. Also, $\mathrm{rank}(p) \neq 1$ for any $p \in L_c$, so the result trivially holds in this case. Suppose the result holds for all terms in L_c of rank at most m, and let $p \in L_c$ have rank $m+1$. If $p = \mu x.q$ for some $q \in L\{X\}$, then since $p \in L_c$, it follows that $q \in L_{\{x\}}$. Now, $\pi_0(p) = b_L \in L_c$ by definition of π_0. Assume that $\pi_n(p) \in L_c$ for some $n \in \mathbb{N}$. Then $\pi_{n+1}(p) = [\pi_n(p)/x](\pi_n(q))$, and the fact that $q \in L_{\{x\}}$ together with $\pi_n(p) \in L$ imply $[\pi_n(p)/x](\pi_n(q)) \in L$ as well. Hence $\pi_n(p) \in L$ for all $n \in \mathbb{N}$ by induction.

Finally, we assume that $p = \omega_{L_c}\langle p_1, \ldots, p_r\rangle$ for some $r \in \mathbb{N}$. Then $p \in L_c$ implies that $\emptyset = \mathrm{Free}(p) = \mathrm{Free}(p_1) \cup \cdots \cup \mathrm{Free}(p_r)$, so $p_i \in L_c$ for each $i = 1, \ldots, r$. Then, for each $n \in \mathbb{N}$, $\pi_n(p) = \omega_{L[X]}\langle \pi_n(p_1), \ldots, \pi_n(p_r)\rangle \in L$ since $\pi_n(p_i) \in L$ by the inductive hypothesis. $\square$

We also need the $\Omega\{X\}$-algebra $L\{X\}[*]$ of contexts over $L\{X\}$. To simplify matters, we assume $* \notin X$. It is necessary to extend the $\Omega\{X\}$-algebra homomorphism Free to Free* whose domain is all of $L\{X\}[*]$ in the following manner. $\mathscr{P}_{\mathrm{fin}}(X \cup \{*\})$ can be made into an $\Omega\{X\}$-algebra in exactly the same way that $\mathscr{P}_{\mathrm{fin}}(X)$ was made into an $\Omega\{X\}$-algebra. Hence, there exists a unique $\Omega\{X\}$-algebra homomorphism $\mathrm{Free}^* : L\{X\}[*] \to \mathscr{P}_{\mathrm{fin}}(X \cup \{*\})$ such that the diagram

$$\begin{tikzcd}
L\{X\} \arrow[r, "incl"] \arrow[d, "\mathrm{Free}"'] & L\{X\}[*] \arrow[d, "\mathrm{Free}^*"'] & \{*\} \arrow[l, "incl"'] \arrow[dl, "* \mapsto \{*\}"] \\
\mathscr{P}_{\mathrm{fin}}(X) \arrow[r, "incl"'] & \mathscr{P}_{\mathrm{fin}}(X \cup \{*\}) &
\end{tikzcd}$$

commutes.

For the following technical lemma, we introduce the Ω-algebra $L[X][*]$ of contexts over $L[X]$, which is a subset of the domain of Free* because it an Ω-subalgebra of $L\{X\}[*]$.

Lemma 2.19. *Let A be an Ω-algebra, $a \in A$, and $C \in L[X][*]$. If $f, f' : L[X] \to A$ are Ω-algebra homomorphisms such that $f|_{\mathrm{Free}^*(C)} = f'|_{\mathrm{Free}^*(C)}$, then $f[* \mapsto a]\,C = f'[* \mapsto a]\,C$.*

Proof. Recalling our assumption that $* \notin X$, we can identify $L[X][*]$ with the polynomial algebra $L[X \cup \{*\}]$, and so we have a rank homomorphism $\rho : L[X][*] \to \mathbb{N}$. The proof then is an easy induction on ρC. $\square$

Now we can prove the important fact that the π_n's are monotone. This is not just a simple consequence of Corollary 2.4 because $L\{X\}$ is regarded as an $\Omega\{X\}$-algebra and $L[X]$ as an Ω-algebra.

Proposition 2.20. *Let $n \in \mathbb{N}$. For all $p, q \in L\{X\}$, $p \preceq_{L\{X\}} q$ implies that $\pi_n(p) \preceq_{L[X]} \pi_n(q)$.*

Proof. The proof is by induction on n.

Suppose $n = 0$ and $p \preceq_{L\{X\}} q$. Let $\phi : L[X] \to A$ be an arbitrary Ω-algebra homomorphism into an ordered Ω-algebra A. Extend the Ω-algebra structure on A to an $\Omega\{X\}$-algebra structure by letting $x_A = \phi(x)$ and $\mu x_A.a = b_A$ for every $x \in X$ and $a \in A$. It is easy to see that $\phi \circ \pi_0 : L\{X\} \to A$ is an $\Omega\{X\}$-algebra homomorphism. By Proposition 2.3, $\phi(\pi_0(p)) \leqslant_A \phi(\pi_0(q))$. Since ϕ is arbitrary, $\pi_0(p) \preceq_{L[X]} \pi_0(q)$.

Suppose that the proposition is true for $n \leqslant k$, and consider the case $n = k + 1$. Assume $p \preceq_{L\{X\}} q$. Now we use induction on the rank of p to show that $\pi_{k+1}(p) \preceq_{L[X]} \pi_{k+1}(q)$.

Suppose the rank of p is 0, i.e., that $p \in L$. Then

$$\pi_{k+1}(p) = \pi_k(p) \preceq_{L[X]} \pi_k(q) \preceq_{L[X]} \pi_{k+1}(q)$$

by the induction hypothesis on n and by Proposition 2.16.

Now suppose we know that $\pi_{k+1}(p) \preceq_{L[X]} \pi_{k+1}(q)$ when the rank$(p) \leqslant j$. Consider the case rank$(p) = j + 1$. We turn to Proposition 2.14 to see the possibilities. We leave the arguments to the reader in all cases save the hardest one: $p = \mu x_{L\{X\}}.p'$ and $q = \mu x_{L\{X\}}.q'$, where $x \in X$ and $p' \preceq_{L\{X\}} q'$. We can identify the Ω-algebra $L[X][*]$ of contexts over $L[X]$ with the Ω-algebra $L[*][X]$ of polynomials in X over $L[*]$. So we have a polynomial substitution homomorphism $[*/x] : L[*][X] \to L[*][X]$, and we have $\mathbf{1}_{L[X]}[* \mapsto x] : L[X][*] \to L[X]$, the substitution of x for $*$ in a context. The restriction to $L[X]$ of the composition $\mathbf{1}_{L[X]}[* \mapsto x] \circ [*/x] : L[X][*] \to L[X]$ is readily seen to be the identity on $L[X]$. Let the context $C \in L[X][*]$ be $C = [*/x](\pi_k(q'))$. Since $\pi_k(q') \in L[X]$,

$$\mathbf{1}_{L[X]}[* \mapsto x]\,C = (\mathbf{1}_{L[X]}[* \mapsto x] \circ [*/x])(\pi_k(q')) = \pi_k(q'). \tag{2}$$

It is easy to see that $x \notin \mathrm{Free}^*(C)$. Let $\phi : L[X] \to A$ be an arbitrary Ω-algebra homomorphism into an ordered Ω-algebra A. Now, we can compute:

$$
\begin{aligned}
\phi(\pi_{k+1}(p)) &= (\phi \circ [\pi_k(p)/x])(\pi_k(p')) \\
&\leqslant_A (\phi \circ [\pi_k(p)/x])(\pi_k(q')) \\
&= (\phi \circ [\pi_k(p)/x])[\mathbf{1}_{L[X]}[* \mapsto x]\, C \\
&= (\phi \circ [\pi_k(p)/x])[* \mapsto \phi(\pi_k(p))]\, C \\
&= \phi[* \mapsto \phi(\pi_k(p))]\, C \\
&\leqslant_A \phi[* \mapsto \phi(\pi_k(q))]\, C \\
&= \cdots \\
&= (\phi \circ [\pi_k(q)/x])(\pi_k(q')) \\
&= \phi(\pi_{k+1}(q)),
\end{aligned}
$$

where the sequence of steps before the $\cdots$ have the following justifications:

(i) By the definition of π_{k+1}.

(ii) By Proposition 2.3 applied to the Ω-algebra homomorphism $\phi \circ [\pi_k(p)/x] : L[X] \to A$, and the induction hypothesis on n that implies that $\pi_k(p') \preceq_{L\{X\}} \pi_k(q')$.

(iii) By equation (2) above.

(iv) By Corollary 2.12.

(v) By Lemma 2.19, which applies because $(\phi \circ [\pi_k(p)/x])|_{\mathrm{Free}^*(C)} = \phi|_{\mathrm{Free}^*(C)}$.

(vi) By Proposition 2.9 and the induction hypothesis on n.

Since ϕ was arbitrary, $\pi_{k+1}(p) \leqslant_A \pi_{k+1}(q)$. This finishes the proof. □

Example 2.21 (Continued). We continue our discussion of the example language S. Using the set X of identifiers, we can construct

(i) $S[X]$, the polynomials over S generated by X,

(ii) $S\{X\}$, the extension of S supporting recursion, but whose terms may contain free occurrences of identifiers, and

(iii) S_c, the extension of S supporting recursion consisting only of closed terms.

Let $x, y \in X$, let $p = \mu x.((b_S\,;x);x)$, and $q = \mu x.((y\,;x);x)$, so that $p \in S_c$ and $q \in S\{X\}$. Since $b_S \preceq_{S[X]} y$, $p \preceq_{S\{X\}} q$. Also,

$$\pi_0(p) = b_S, \qquad \pi_1(p) = b_S\,;b_S\,;b_S, \qquad \pi_2(p) = b_S\,;b_S\,;b_S\,;b_S\,;b_S\,;b_S\,;b_S$$

and

$$\pi_0(q) = b_S, \qquad \pi_1(q) = (y\,;b_S);b_S, \qquad \pi_2(p) = (y\,;((y\,;b_S);b_S));((y\,;b_S);b_S).$$

A point to appreciate here is that, whereas ; is associative in S, our constructions of $S[X]$ and $S\{X\}$ do not make ; associative in these algebras – hence the need for explicit parentheses in some of the expressions above. We will address the issue of requiring equations to hold in extensions of languages in the last section. □

3. Operational and denotational models

The results of the previous section apply to universal algebras quite broadly, and are not confined to the realm of programming languages. We now are ready to bring more semantics issues to the fore, and we begin by recalling the role of operational and denotational models for programming languages. We start with the denotational side.

If L is a programming language viewed as an Ω-algebra, then a denotational model for L is simply an Ω-algebra D and a homomorphism $\mathscr{D} : L \to D$. If we assume that L has no identifiers or recursive constructs, then any Ω-algebra would do for D. Of course, any Ω-algebra D must have an element b_D because of our standing assumption about the constant $b \in \Omega$.

We now give an explanation of the operational and denotational models for the language L, and the relationship between them. If O is the set of *observations* we can make of programs in L, then the *operational semantics* of L is some fixed *behavior function* $\mathscr{O} : L \to O$. Typically, the behavior function is defined using a transition system or rewriting system, but where it comes from is not the issue here.

The usual definitions of adequacy and full abstraction can be stated as follows. The denotational model $\mathscr{D}$ is said to be *adequate* for the operational model $\mathscr{O}$ if $\mathscr{D}(p) = \mathscr{D}(q)$ implies that $\mathscr{O}(p) = \mathscr{O}(q)$ for every $p, q \in L$. In addition, $\mathscr{D}$ is *fully abstract* with respect to $\mathscr{O}$ if, given $p, q \in L$ such that $\mathscr{D}(p) \neq \mathscr{D}(q)$, there exists some context $C \in L[*]$ such that $\mathscr{O}(C[p]) \neq \mathscr{O}(C[q])$. We now proceed to augment these definitions to suit the setting we are in.

Definition 3.1. $\mathscr{O} : L \to O$ is an *ordered operational model* for L if O is a poset and $\mathscr{O}$ is monotone with respect to $\preceq_L$, the minimum monotone pre-order on L. Likewise, the denotational model $\mathscr{D} : L \to D$ is an *ordered denotational model* for L if D is a poset that also is an Ω-algebra in which all operators of Ω have monotone interpretations.

If $\mathscr{O} : L \to O$ is an ordered operational model and $\mathscr{D} : L \to D$ is an ordered denotational model, then we say

(i) $\mathscr{D}$ is *order adequate* with respect to $\mathscr{O}$ if $\mathscr{D}(p) \leqslant_D \mathscr{D}(q)$ implies that $\mathscr{O}(p) \leqslant_O \mathscr{O}(q)$ for all $p, q \in L$.

(ii) $\mathscr{D}$ is *order fully abstract* with respect to $\mathscr{O}$ if given $p, q \in L$ with $\mathscr{D}(p) \nleqslant_D \mathscr{D}(q)$, there is some context $C \in L[*]$ such that $\mathscr{O}(C[p]) \nleqslant_O \mathscr{O}(C[q])$.

We can give an alternative characterization of order adequacy when the models have more structure. If D is a poset and $K(D)$ denotes the set of compact elements of D, then D is an *algebraic poset* if $K(d) = \{k \in K(D) \mid k \leqslant d\}$ is directed and satisfies $d = \bigsqcup K(d)$ for each $d \in D$. The point is that D need not be a cpo.

Proposition 3.2. *Suppose $\mathscr{O} : L \to O$ is an ordered operational model for L and $\mathscr{D} : L \to D$ is an ordered denotational model such that O is a cpo, D is an algebraic* *poset, and $\mathscr{D}(L) = K(D)$. The following are equivalent:*

(i) *$\mathscr{D}$ is order adequate with respect to $\mathscr{O}$.*

(ii) *There is a Scott continuous map $B_D : D \to O$ such that $\mathscr{O} = B_D \circ \mathscr{D}$.*

Proof. Suppose $\mathscr{D}$ is order adequate with respect to $\mathscr{O}$. For any $d \in D$, we claim that $\{\mathscr{O}(p) \mid \mathscr{D}(p) \leqslant_D d\}$ is a directed subset of O. Note that since $\mathscr{D}(L) = K(D)$, the family $\downarrow d \cap \mathscr{D}(L) = K(d)$ is directed in D. Indeed, if $p, p' \in L$ satisfy $\mathscr{D}(p), \mathscr{D}(p') \leqslant_D d$, then there is some compact element of D, and hence some element $q \in L$ such that $\mathscr{D}(p), \mathscr{D}(p') \leqslant_D \mathscr{D}(q) \leqslant_D d$. The claim that $\{\mathscr{O}(p) \mid \mathscr{D}(p) \leqslant_D d\}$ is directed now follows from order adequacy. Thus, we can define

$$B_D : D \to O \quad \text{by} \quad B_D(d) = \bigsqcup \{\mathscr{O}(p) \mid \mathscr{D}(p) \leqslant_D d\}.$$

From this definition, we see that $\mathscr{O}(p) = \bigsqcup\{\mathscr{O}(q) \mid \mathscr{D}(q) \leqslant_D \mathscr{D}(p)\} = B_D(\mathscr{D}(p))$. Hence $\mathscr{O} = B_D \circ \mathscr{D}$. Also, if $k, k' \in K(D)$, then $k = \mathscr{D}(p)$ and $k' = \mathscr{D}(p')$ for some $p, p' \in L$, and so $k \leqslant_D k'$ implies that $\mathscr{D}(p) \leqslant_D \mathscr{D}(p')$, which in turn implies that $\mathscr{O}(p) \leqslant_O \mathscr{O}(p')$. Since $B_D(k) = \mathscr{O}(p)$ and $B_D(k') = \mathscr{O}(p')$, it follows that $B_D|_{K(D)}$ is monotone. Since D is an algebraic poset, it then follows that $B_D|_{K(D)}$ has a unique Scott-continuous extension B to all of D which is given by $B(d) = \bigsqcup B_D(K(d))$. Thus,

$$\begin{aligned} B(d) = \bigsqcup B_D(K(d)) &= \bigsqcup \{B_D \circ \mathscr{D}(p) \mid \mathscr{D}(p) \leqslant_D d\} \\ &= \bigsqcup \{\mathscr{O}(p) \mid \mathscr{D}(p) \leqslant_D d\} = B_D(d), \end{aligned}$$

the second equality following since $\mathscr{D}(L) = K(D)$, and the last by the definition of B_D.

The converse is clear from the monotonicity of Scott continuous maps. $\square$

The next result is important, but obvious.

Proposition 3.3. (i) *If $\mathscr{D}$ is order adequate with respect to $\mathscr{O}$, then $\mathscr{D}$ is adequate with respect to $\mathscr{O}$.*

(ii) *If $\mathscr{D}$ is order fully abstract with respect to $\mathscr{O}$, then $\mathscr{D}$ is fully abstract with respect to $\mathscr{O}$.*

Proposition 3.4. *If $\mathscr{O} : L \to O$ is a ordered operational model for L and $\mathscr{D} : L \to D$ is an ordered denotational model for L, then the following are equivalent:*

(i) *$\mathscr{D}$ is order adequate and order fully abstract with respect to $\mathscr{O}$.*

(ii) *For all $p, q \in L$, $\mathscr{D}(p) \leqslant_D \mathscr{D}(q)$ if and only if $\mathscr{O}(C[p]) \leqslant_O \mathscr{O}(C[q])$ for all contexts $C \in L[*]$.*

Proof. Suppose $\mathscr{D}$ is order adequate with respect to $\mathscr{O}$. Assume $\mathscr{D}(p) \leqslant_D \mathscr{D}(q)$, and let $C \in L[*]$. By Corollary 2.12 and Proposition 2.9,

$$\begin{aligned} \mathscr{D}(C[p]) = \mathscr{D}(\mathbf{1}_L[* \mapsto p]\, C) &= \mathscr{D}[* \mapsto \mathscr{D}(p)]\, C \\ &\leqslant_D \mathscr{D}[* \mapsto \mathscr{D}(q)]\, C = \mathscr{D}(C[q]). \end{aligned}$$

Now order adequacy gives $\mathscr{O}(C[p]) \leqslant_O \mathscr{O}(C[q])$. The rest is clear. $\square$

Remark. It is important to note that the proof of the last proposition depends on the fact that substitution of $p \in L$ in a context $C \in L[*]$ is defined by $C[p] = \mathbf{1}_L[* \mapsto p]\,C$, which is appropriate to the setting of a language without binding operators. The proof of the analogous result for the extended language L_c of closed terms (cf. Section 6) requires substantially different arguments, because of the relationship between substitution and binding operators.

Our goal is to find conditions that ensure that, given an ordered operational model $\mathcal{O} : L \to O$ and a related ordered denotational model $\mathcal{D} : L \to D$ which is order adequate and fully abstract with respect to $\mathcal{O}$, these models can be extended in a natural way to models $\mathcal{O}_c : L_c \to O_c$ and $\mathcal{D}_c : L_c \to D_c$ so that $\mathcal{D}_c$ is order adequate and fully abstract with respect to $\mathcal{O}_c$. The key is to assume a strengthened form of order full abstraction that says that, whenever p can be distinguished operationally from q, there is a single context that distinguishes p from every program that q approximates.

Definition 3.5. Let $\mathcal{O} : L \to O$ be an ordered operational model and $\mathcal{D} : L \to D$ is an ordered denotational model for L as defined above. Then we say

- $\mathcal{D}$ is *strongly order fully abstract* with respect to $\mathcal{O}$ if, given $p, q \in L$ with $\mathcal{D}(p) \not\leq_D \mathcal{D}(q)$, there is some context $C \in L[*]$ satisfying $\mathcal{O}(C[p]) \not\leq_O \mathcal{O}(C[q'])$ in O for every $q' \in L$ with $q \preceq_L q'$.

Note that this notion is stronger than the notion of order full abstraction we gave above, and hence it also is stronger than the “usual” notion of full abstraction.

Example 3.6 (Continued). We continue our discussion of the example language S we defined in Section 2. Recall that S has the following set of BNF-like production rules:

$$p ::= b_S \mid \delta_S \mid \varepsilon_S \mid a \mid p\,;q \mid p+q,$$

and we view S as the free (Σ, E)-algebra, where Σ is the signature consisting of constants b_S, δ_S, and ε_S, the constants $a \in A$, and binary operators ; and $+$.

We now define an operational model for S. First, let $\surd \notin A$ be introduced as an observable signal that a program has terminated normally. An *atomic transition* is a ordered triple written $p \xrightarrow{x} p'$, where $p, p' \in S$ and $x \in A \cup \{\surd\}$, signifying that execution of the next step of a program p can result in the observable action x with the remainder left to be executed described by p'. Next, we introduce the following system of transition rules, where $x \in A \cup \{\surd\}$:

(i) $\varepsilon \xrightarrow{\surd} \delta$

(ii) $a \xrightarrow{a} \varepsilon$

(iii) $a\,;p \xrightarrow{a} p$

(iv) $\dfrac{p_1 \xrightarrow{x} p_2}{p_1\,;m \xrightarrow{x} p_2\,;m}$

(v) $\dfrac{p_1 \xrightarrow{x} p_2}{p_1 + m \xrightarrow{x} p_2}$

From this transition system we obtain a function

$$B : S \rightarrow \{\{\delta\}\} \cup \mathscr{P}_{\mathrm{nf}}(A^* \cup A^+\delta \cup A^*\surd),$$

where A^* denotes the finite words over the alphabet A, A^+ denotes the finite, nonempty words over A, and $\mathscr{P}_{\mathrm{nf}}(X)$ denotes the family of finite, nonempty subsets of the set X. In more detail, we define

$$B\,p = \{s \mid s \in A^* \ \& \ p \xrightarrow{s} b_S\,; p'\} \cup \{s\delta \mid s \in A^* \ \& \ l \xrightarrow{s} \delta\}$$
$$\cup \ \{s\surd \mid s \in A^* \ \& \ p \xrightarrow{s} p' \xrightarrow{\surd} \delta\},$$

where the notation $p \xrightarrow{s} p'$ means there is a composable sequence of atomic transitions starting with p and ending with p', the concatenation of whose actions is s.

We now show that $D_0 = \{\{\delta\}\} \cup \mathscr{P}_{\mathrm{nf}}(A^* \cup A^+\delta \cup A^*\surd)$ also is a (Σ, E)-algebra. Let λ denote the empty string. We define the constants by $b_{D_0} = \{\lambda\}$, $\delta_{D_0} = \{\delta\}$, and $\varepsilon_{D_0} = \{\surd\}$. We define the constants a in D_0 by $a_{D_0} = \{a\}$ for each $a \in A$. To define the operation ; on D_0, we first define a related operation $\hat{;}$ on $A_{\mathrm{fin}} = A^* \cup A^*\delta \cup A^*\surd$ by

$$s \,\hat{;}\, t = \begin{cases} s & \text{if} s \notin A^*\surd, \\ s'\,t & \text{if} s = s'\surd \in A^*\surd. \end{cases}$$

It is routine to show that this operation makes A_{fin} a monoid with identity $\surd$ and left-zero λ. We then define the operation ; : $D_0 \times D_0 \rightarrow D_0$ by $F;G = \{x \,\hat{;}\, y \mid x \in \ F \ \& \ y \in G\}$. Again, this makes D_0 into a monoid, with $\{\surd\}$ the identity and $\{\delta\}$ a left-zero. Finally, we define the operation $+$ on D_0 as follows:

$$F + G = \begin{cases} F & \text{if} G = \{\delta\}, \\ G & \text{if} F = \{\delta\}, \\ F \cup G & \text{otherwise}. \end{cases}$$

Thus, $+$ is the operation of union, modified so that $\{\delta\}$ is a $+$-identity. Defined in this way $+$ models "external nondeterminism," whereby a process that is able to perform an a action in response to a request from an external world, in fact will perform an a. In part because our language does not include any form of synchronization (something that is omitted for simplicity's sake), we cannot further illustrate this aspect of $+$.

With operations defined in this manner, it is routine to show that D_0 is an (Σ, E)-algebra. We also want to impose a monotone pre-order on D_0 and we define this order using the results of the previous section.

The prefix order $\sqsubseteq$ on the set $A^* \cup A^*\delta \cup A^*\surd$ enables us to define the Egli–Milner preorder on D_0 :

$$F \preceq_P Q \quad \text{if and only if} \quad F \subseteq {\downarrow} G \ \& \ G \subseteq {\uparrow} F,$$

where ${\downarrow} G = \{s \in A_{\mathrm{fin}} \mid (\exists t \in G)\ s \sqsubseteq t\}$, and ${\uparrow} F = \{s \in A_{\mathrm{fin}} \mid (\exists t \in F)\ t \sqsubseteq s\}$.

Theorem 3.7. *The minimum Σ-monotone preorder on D_0 is the same as the Egli-Milner preorder on D_0.*

Proof. It is a straightforward verification that the Egli–Milner pre-order is a monotone pre-order on D_0, and so the minimum monotone pre-order is a subset of this pre-order.

To show the converse, we first note that, given $s,t \in A_{\text{fin}}$ with $s \sqsubseteq t$, either $s = t$ or else there is some $x \in A_{\text{fin}}$ and some $s' \in A^*\surd$ so that $s = s';\lambda$ and $t = s';x$. Since λ is the least element of A_{fin}, it follows that $\phi(s) = \phi(s';\lambda) = \phi(s');\phi(\lambda) \leqslant_C \phi(s');\phi(x) = \phi(t)$ for any ;-homomorphism $\phi : A_{\text{fin}} \to C$ into an ordered ;-algebra. Proposition 2.3 then implies that the prefix order on A_{fin} is the minimum monotone pre-order on A_{fin} regarded as a ;-algebra.

Next, if $\phi : D_0 \to C$ is a Σ-algebra homomorphism into an ordered (Σ,E)-algebra, then $s \mapsto \phi(\{s\})$ is an ;-algebra map, and so $\phi \circ \{-\}$ preserves the prefix order on A_{fin}. Now, if $F,G \in D_0$ satisfy $F \preceq_P Q$, then $F = \{s_1,\ldots,s_m\}$, $G = \{t_1,\ldots,t_n\}$ and $(\forall i)(\exists j)\ s_i \sqsubseteq t_j$ and $(\forall j)(\exists i)\ s_i \sqsubseteq t_j$. It follows that

$$\begin{aligned}\phi(F) = \phi(\{s_1\} + \cdots + \{s_m\}) &= \phi(\{s_1\}) + \cdots + \phi(\{s_m\})\\ &\leqslant_C \phi(\{t_1\}) + \cdots + \phi(\{t_n\}) = \phi(G).\end{aligned}$$

Hence $F \preceq_P G$ implies that $F \preceq_{D_0} G$. $\square$

Now, the (Σ,E)-algebra D_0 has a quotient algebra D_1 which is an ordered (Σ,E) algebra – namely, we take $D_1 = D_0/\equiv$, where $\equiv = \preceq_{D_0} \cap (\preceq_{D_0})^{-1}$, the equivalence relation induced by the pre-order $\preceq_{D_0}$. This algebra is partially ordered by the order $\preceq_{D_0}$ induces on it. However, D_1 is not a cpo, and so we let D be the ideal completion of $(D_1, \preceq_{D_0}/\equiv)$ (cf., [10]); it is easy to show that each of the operations of Σ extends to a continuous operation on D. So, D is a continuous (Σ,E)-algebra. If $[F]$ denotes the ideal the equivalence class of F generates in the ideal completion of D_1, then we know there is a Σ-algebra homomorphism of $\mathscr{D} : S \to D$, and it satisfies

- $\mathscr{D}(b_S) = [\{\lambda\}]$,
- $\mathscr{D}(\delta_S) = [\{\delta\}]$, and
- $\mathscr{D}(\varepsilon_S) = [\{\varepsilon\}]$,

since S is the initial ordered (Σ,E)-algebra. Thus, we define our denotational model to be the cpo D; note that $\mathscr{D}(S) = K(D)$, since it is easy to construct a term in S whose image under $\mathscr{D}$ is any given element of D_0.

We also define our ordered operational model $O = D$, and we define a *behavior map* $\mathscr{O} : S \to O$ by $\mathscr{O}(s) = [B(s)]$, where $B : S \to D_0$ is the mapping we defined above using the transition system. Then both O and D are cpo's, and our goal is to show that the mappings $\mathscr{O}$ and $\mathscr{D}$ are identical.

Theorem 3.8. *The mappings $\mathscr{D} : S \to D$ and $\mathscr{O} : S \to O$ are equal.*

Proof. Let S_0 be the initial Σ-algebra. Then S is a quotient of S_0. The advantage is that S_0 is an initial anomic algebra, and so we can apply structural induction here. This is not so clearly possible for S, since there are equations in S which have to be dealt with. In any case, since S_0 is an initial algebra of the same signature as D, there is a Σ-algebra homomorphism $\Phi : S_0 \to D$ satisfying the clauses enumerated above for $\mathscr{D}$. Since S is the quotient of S_0 modulo the congruence generated by the equations in E, it follows that Φ factors through the quotient mapping from S_0 to S. That is, if $\pi : S_0 \to S$ is the quotient mapping, then there is a Σ-algebra map $\psi : S \to D$ satisfying $\Phi = \psi \circ \pi$. Since $\mathscr{D}$ is the unique Σ-algebra map from S to D, we have $\psi = \mathscr{D}$.

Also, we can define a mapping $\mathscr{O}_0 : S \to D$ by composing the quotient map $\pi : S_0 \to S$ with the mapping $\mathscr{O} : S \to O$. Clearly, then, we will be finished if we show that $\mathscr{O}_0 = \Phi$. A routine structural induction on the clauses defining the elements of S_0 shows that $\mathscr{O}_0$ is a Σ-algebra map. For example, the clauses enumerated above that Φ and $\mathscr{D}$ satisfy assure that $\mathscr{O}_0$ acts correctly on the constants of the language S_0. And, if l and m are terms of S_0, then an induction argument on the rank of the terms (where we use the algebraic rank function which assigns 0 to constants and adds 1 for each operator) shows that $\mathscr{O}_0(l);\mathscr{O}_0(m) = \mathscr{O}_0(l;m)$ and $\mathscr{O}_0(l)+\mathscr{O}_0(m) = \mathscr{O}_0(l+m)$. Since Φ is the unique Σ-algebra map from S_0 to D, it follows that $\Phi = \mathscr{O}_0$. □

Remark 3.9. Note the fact that the assignment $p \mapsto \mathscr{O}\, p$ defined on S is a function follows from this corollary. Indeed, by showing that $\Phi = \mathscr{O}_0$, we have concluded that $\mathscr{O} = \mathscr{D}$ as well. This implies that $\mathscr{O}$ also is a function. The point is that since S is an initial algebra subject to the equations E, we cannot apply structural induction to S directly, and so a proof about terms of S either has to rely on deriving some normal form for those terms, or else on arguments that take the many forms a term can have into account. Either approach ultimately relies on the fact that S is a quotient of S_0, an initial anomic algebra to which structural induction applies.

Corollary 3.10. *The denotational model $\mathscr{D} : S \to D$ is order adequate and strongly order fully abstract with respect to the operational model $\mathscr{O} : S \to O = D$.*

Proof. Since $\mathscr{D} = \mathscr{O}$, if $\mathscr{D}(s) \not\leq_D \mathscr{D}(t)$, then we can take C to be the identity context $*$, and $\mathscr{O}(C[s]) \not\leq_O \mathscr{O}(C[t])$. □

Thus we have the models for the finitary language S that we need to apply our theory. In the ensuing sections, we will delineate exactly how the denotational model D can be extended to one for the language of closed terms of the extension of the language S to include identifiers and recursion operators, and how that model will relate to the operational $O = D$.

4. Denotational semantics of L_c

This section deals exclusively with the construction of denotational models of extensions of L. Throughout this section the only assumptions we will make are that $\mathscr{D} : L \to D$ is an ordered denotational model for L and that D is a continuous Ω-algebra. From $\mathscr{D}$ we will construct in succession the homomorphisms $\mathscr{D}_{L[X]}$, $\mathscr{D}_{L\{X\}}$, and $\mathscr{D}_c$ that give the semantics of $L[X]$, $L\{X\}$, and L_c, respectively. Some of the results in this section are generalizations of results proved for particular languages. For example, see [3, Ch. 7].

Let $[D^X \to D]$ be the collection of *Scott continuous* maps from the algebraic cpo of semantic environments to D, ordered pointwise. Since D is a continuous Ω-algebra, $[D^X \to D]$ is also a continuous Ω-algebra, with operations defined pointwise. By ignoring the semantic environment, it is easy to get a continuous homomorphism $\kappa_D : D \to [D^X \to D]$ given by $\kappa_D dh = d$, where $d \in D$ and $h \in D^X$. Let $\upsilon_D : X \to [D^X \to D]$ be the evaluation function $\upsilon_D xh = hx$, where $x \in X$ and $h \in D^X$. Although there is an obvious difficulty in assigning terms in $L[X]$ denotational meanings in D, there is no problem in making $[D^X \to D]$ the target of a semantic homomorphism by letting

$$\mathscr{D}_{L[X]} : L[X] \to [D^X \to D]$$

be the unique Ω-algebra homomorphism that makes

$$\begin{tikzcd} L \arrow[r, "\iota"] \arrow[d, "\mathscr{D}"'] & L[X] \arrow[d, "\mathscr{D}_{L[X]}"'] & X \arrow[l, "incl"'] \arrow[dl, "\upsilon_D"] \\ D \arrow[r, "\kappa_D"] & {[D^X \to D]} & \end{tikzcd}$$

a commutative diagram.

We want $[D^X \to D]$ to be an ordered model, not only of $L[X]$, but also of $L\{X\}$. Hence, we want to show there is a canonical $\Omega\{X\}$-algebra homomorphism $\mathscr{D}_{L\{X\}} : L\{X\} \to [D^X \to D]$. To do this, we must first make $[D^X \to D]$ a continuous $\Omega\{X\}$-algebra by defining interpretations in $[D^X \to D]$ of the appropriate operators and constants.

One way in which the signature $\Omega\{X\}$ extends the signature Ω is that each $x \in X$ becomes a 0-ary operator. Since we want our semantics for $L\{X\}$ to extend the semantics given above for its Ω-subalgebra $L[X]$, we let

$$x_{[D^X \to D]} = \upsilon_D x = \lambda h.hx.$$

The other aspect of the extension of Ω to $\Omega\{X\}$ relates to the addition of the recursion operators. Thus, we must define continuous interpretations of the unary operators $\mu x.-$ for $x \in X$. Their definition begins with the observation that a semantic environment can be "locally overridden" by altering the value associated with a single identifier. Thus, if $x \in X$, $h \in D^X$, and $d \in D$, then the "locally overridden" semantic environment $[h \mid x \mapsto d] \in D^X$ is given by

$$[h \mid x \mapsto d]y = \begin{cases} d & \text{if } y = x, \\ hy & \text{otherwise.} \end{cases}$$

We can then define the mapping $p_x : D^X \times D \to D^X$ by $p_x(h,d) = [h \mid x \mapsto d]$. To see in an easy fashion that p_x is Scott continuous, imagine the product object $D^X \times D$ to be "rearranged" so that p_x coincides with a projection of the product object onto a factor, and use the fact that such projections are continuous. The fact that the category of cpo's with Scott continuous maps is cartesian closed then implies that, for any $f \in [D^X \to D]$, there is a Scott continuous map $\mathrm{cur}(f \circ p_x) : D^X \to [D \to D]$ such that the following diagram commutes:

$$\begin{tikzcd} D^X \times D \arrow[r, "\mathrm{cur}(f\circ p_x)\times 1_D"] \arrow[d, "p_x"'] & {[D \to D] \times D} \arrow[d, "\mathrm{ap}"] \\ D^X \arrow[r, "f"] & D \end{tikzcd}$$

where the Scott continuous map $\mathrm{ap} : [D \to D] \times D \to D$ is application of first argument to the second. In particular, if $h \in D^X$ and $d \in D$, then

$$\mathrm{cur}(f \circ p_x)hd = f[h \mid x \mapsto d].$$

Note that the mapping

$$f \mapsto \mathrm{cur}(f \circ p_x) : [D^X \to D] \to [D^X \times D \to [D \to D]]$$

is Scott continuous. Since D is a cpo, by the Scott–Tarski–Knaster Theorem, there is a Scott continuous least fixed point operator $\mathrm{Y}_D : [D \to D] \to D$ that assigns to each continuous self-map of D its least fixed point. Combining Y_D with the mapping $\mathrm{cur}(f \circ p_x) : D^X \to [D \to D]$, we obtain the desired continuous interpretation of the recursion operator:

$$\mu x_{[D^X \to D]}.f \;=\; \mathrm{Y}_D \circ \mathrm{cur}(f \circ p_x) \in [D^X \to D].$$

The next proposition clarifies the character of the recursion operator $\mu x_{[D^X \to D]}$.

Proposition 4.1. *If $x \in X$, $f \in [D^X \to D]$ and $h \in D^X$, then*

$$\mu x_{[D^X \to D]}.fh$$

is the least fixed point of the mapping

$$d \mapsto \mathrm{cur}(f \circ p_x)hd = f[h \mid x \mapsto d] \in [D \to D].$$

Proof. From the definition of $\mu x_{[D^X \to D]}$, we see that

$$\mu x_{[D^X \to D]}.fh = \mathrm{Y}_D(\mathrm{cur}(f \circ p_x)h). \qquad \square$$

Since $[D^X \to D]$ is now a continuous $\Omega\{X\}$-algebra and $\kappa_D \circ \mathscr{D} : L \to [D^X \to D]$ is an Ω-algebra homomorphism, the universal property of $L\{X\}$ implies there is a unique $\Omega\{X\}$-algebra map $\mathscr{D}_{L\{X\}} : L\{X\} \to [D^X \to D]$ extending $\kappa_D \circ \mathscr{D}$:

$$\begin{tikzcd} L \arrow[r] \arrow[d, "\mathscr{D}"'] & L\{x\} \arrow[d, "\mathscr{D}_{L\{X\}}"] \\ D \arrow[r, "\kappa_D"'] & {[D^X \to D]} \end{tikzcd}$$

By Proposition 2.3, since $[D^X \to D]$ is an ordered $\Omega\{X\}$-algebra, the mapping $\mathscr{D}_{L\{X\}}$ is monotone with respect to $\preceq_{L\{X\}}$. Note that the restriction of $\mathscr{D}_{L\{X\}}$ to $L[X]$ equals $\mathscr{D}_{L[X]}$ because the former is an Ω-homomorphism that satisfies the universal mapping property that defines the latter,

The next proposition gives some equational properties of $\mathscr{D}_{L\{X\}}$.

Proposition 4.2. *Let $x \in X$, $p, p_1, \ldots, p_n \in L$, and $\omega \in \Omega$. Then*

(i) $\mathscr{D}_{L\{X\}}\, b_{L\{X\}} = \lambda h.\ \bot_D$,

(ii) $\mathscr{D}_{L\{X\}}\, x_{L\{X\}} = \lambda h.(h\,x)$,

(iii) $\mathscr{D}_{L\{X\}}\,(\omega_{L\{X\}}\langle p_1, \ldots, p_n\rangle) = \lambda h.(\omega_D\langle \mathscr{D}_{L\{X\}}\, p_1\, h, \ldots, \mathscr{D}_{L\{X\}}\, p_n\, h\rangle)$,

(iv) $\mathscr{D}_{L\{X\}}\,(\mu_{L\{X\}}.p) = \lambda h.(\mathrm{Y}_D \Phi_{x,p,h})$, *where* $\Phi_{x,p,h} : D \to D$ *is given by*

$$\Phi_{x,p,h} d = \mathscr{D}_{L\{X\}}\; p\,[h \mid x \mapsto d].$$

Proof. This follows easily from the fact that $\mathscr{D}_{L\{X\}} : L\{X\} \to [D^X \to D]$ is an ordered denotational model, so $\mathscr{D}_{L\{X\}}$ both preserves minimal elements and is a homomorphism of $\Omega\{X\}$-algebras. □

In what may be called the "standard approach" (see [3, Ch. 7]) to the denotational semantics of languages supporting recursion, the equations of Proposition 4.2 are taken to be the definition $\mathscr{D}_{L\{X\}}$ via structural induction. Even if our goal were only to provide a definition of $\mathscr{D}_{L\{X\}}$ as a function, the standard approach would not be any simpler than the approach we have given because the standard approach entails, not only giving a definition of $\mathscr{D}_{L\{X\}}$, but also giving simultaneously definitions of all the functions $\Phi_{x,p,h}$ and resolving the ensuing apparent circularity, as well as proving continuity of the functions $\Phi_{x,p,h}$ (so that the Scott–Tarski–Knaster Theorem can be applied to extract fixed points). Our approach has the advantage of additionally making perfectly clear the algebraic character of $\mathscr{D}_{L\{X\}}$ as an $\Omega\{X\}$-homomorphism satisfying a universal mapping property. Besides the technical utility of such an understanding (evidence of which will be found repeatedly in the remainder of this paper), by thinking in algebraic terms we have gained insights necessary to formulate our results in a very general setting.

The following proposition generalizes Lemma 7.14(a) in [3].

Proposition 4.3. *If $p \in L\{X\}$ and $h, h' \in D^X$ satisfy $hx = h'x$ for all $x \in$* Free(p), *then $\mathscr{D}_{L\{X\}}\, p\,h = \mathscr{D}_{L\{X\}}\, p\,h'$.*

Proof. We proceed by induction on rank(p). If rank(p) = 0, then $p \in L$, and so $\mathscr{D}_{L\{X\}}\, p = \kappa_D(\mathscr{D} p)$ is a constant function, so the result follows trivially.

If rank(p) = 1, then $p = x \in X$. Because $\mathscr{D}_{L\{X\}}$ and $\mathscr{D}_{L[X]}$ agree on $L[X]$, $\mathscr{D}_{L\{X\}}\, p\, h = \mathscr{D}_{L[X]}\, x\, h = v_D x h = hx = h'x = \mathscr{D}_{L\{X\}}\, p\, h'$, so that the result holds in this case as well.

Suppose that the result holds for all terms of rank at most $n > 0$, and let $p \in L\{X\}$ have rank $n+1$. If $p = \mu x.p'$ for some $p' \in L\{X\}$ with rank$(p') \leqslant n$, then

$$\begin{aligned}\mathscr{D}_{L\{X\}}\, p\, h = \mathscr{D}_{L\{X\}}\, (\mu x.p')h &= \mu x_{[D^X \to D]}.(\mathscr{D}_{L\{X\}}\, p')h \\ &= \mathrm{Y}_D \left(\mathrm{cur}\,(\mathscr{D}_{L\{X\}}\, p' \circ p_x)\, h\right)\end{aligned}$$

and

$$\begin{aligned}\mathscr{D}_{L\{X\}}\, p\, h' = \mathscr{D}_{L\{X\}}\, (\mu x.p')h' &= \mu x_{[D^X \to D]}.(\mathscr{D}_{L\{X\}}\, p')h' \\ &= \mathrm{Y}_D \left(\mathrm{cur}\,(\mathscr{D}_{L\{X\}}\, p' \circ p_x)\, h'\right).\end{aligned}$$

So, to show that $\mathscr{D}_{L\{X\}}\, p\, h = \mathscr{D}_{L\{X\}}\, p\, h'$, it suffices to show that the functions

$$\mathrm{cur}\,(\mathscr{D}_{L\{X\}}\, p' \circ p_x)\, h : D \to D \quad \text{and} \quad \mathrm{cur}\,(\mathscr{D}_{L\{X\}}\, p' \circ p_x)\, h' : D \to D$$

are the same, since they then will have the same least fixed point. But, for any $d \in D$

$$\begin{aligned}\mathrm{cur}\,(\mathscr{D}_{L\{X\}}\, p' \circ p_x)\, hd &= \mathrm{ap}\left(\mathrm{cur}\,(\mathscr{D}_{L\{X\}}\, p' \circ p_x)\, h, d\right) \\ &= \mathscr{D}_{L\{X\}}\, p'(p_x(h,d)) \\ &= \mathscr{D}_{L\{X\}}\, p'(p_x(h',d)) \\ &= \mathrm{cur}\,(\mathscr{D}_{L\{X\}}\, p' \circ p_x)\, h'd,\end{aligned}$$

the third equality following from the induction hypothesis.

Similarly, if $p = \omega_{L\{X\}}\langle p_1, \ldots, p_r\rangle$, then the facts that $\mathscr{D}_{L\{X\}}$ is an $\Omega\{X\}$-homomorphism and that all the p_i's have lower rank implies the result holds in this case as well. □

Corollary 4.4. *There exists a unique Ω-algebra homomorphism $\mathscr{D}_c : L_c \to D$ such that the diagram*

$$\begin{tikzcd}
L \arrow[rr] \arrow[dr] \arrow[dd, "\mathscr{D}"'] & & L\{X\} \arrow[dd, "\mathscr{D}_{L\{X\}}"] \\
& L_c \arrow[ur] \arrow[dl, "\mathscr{D}_c"'] & \\
D \arrow[rr, "\kappa_D"'] & & {[D^X \to D]}
\end{tikzcd}$$

commutes, where the unlabeled arrows all are inclusions. Hence, $\mathscr{D}_c : L_c \to D$ is an ordered denotational model of L_c. Note also that, for all $p \in L_c$ and all $h \in D^X$,

$$\mathscr{D}_c\, p = \mathscr{D}_{L\{X\}}\, p\, h.$$

Because the terms of the language $L\{X\}$ have binding operators μx in them, we need a more elaborate notion of substitution than is provided by the comparatively simple algebraic approach that was given for $L[X]$ in Section 3. In general, when substitution is defined for languages with binding operators it appears to be in terms of transformations that are not homomorphisms. For instance, suppose $x, y \in X$ and $x \neq y$. Let

$$[y_{L\{X\}}/x] : L\{X\} \to L\{X\}$$

be a function that substitutes y for free occurrences of x in its argument. In whatever manner this function is defined, surely we want

$$[y_{L\{X\}}/x]\, x_{L\{X\}} = y_{L\{X\}}.$$

If $[y_{L\{X\}}/x]$ were a homomorphism of $\Omega\{X\}$-algebras, then we also would have that,

$$[y_{L\{X\}}/x]\,(\mu z_{L\{X\}}.x_{L\{X\}}) = \mu z_{L\{X\}}.([y_{L\{X\}}/x]\, x_{L\{X\}}) = \mu z_{L\{X\}}.\, y_{L\{X\}},$$

for all $z \in X$. If we suppress the subscripts, we can rewrite this more informally as

$$[y/x]\,(\mu z.x) = \mu z.([y/x]\,x) = \mu z.\, y.$$

This last equation is wrong on two counts. First, if $z = x$, we want

$$[y/x]\,(\mu x.x) = \mu x.x,$$

and, second, if $z = y$, we want

$$[y/x]\,(\mu y.x) = \mu w.\, y,$$

where $w \neq y$.[4] The latter problem is that of avoiding capture by a binding operator of identifiers that should occur freely. However, we want – indeed, we need – a treatment of substitution in terms of homomorphisms in order to avoid nonalgebraic *ad hoc* case arguments that result when giving complete proofs based on the Curry–Feys definition of substitution. Luckily, the observations above show only that the most naive approach imaginable does not work. It is still possible to give a treatment of substitution in the presence of binding operators in which the process of substitution is the application of a homomorphism.

The essence of the algebraic approach to substitution has two features:

(i) Treatment from the outset of simultaneous substitution, as opposed to dealing with substitution one identifier at a time. A simultaneous substitution corresponds to a syntactic environment that assigns to each identifier a term of a language.

(ii) Proper identification of the sets that are the carriers of the algebras that are the domain and codomain of the substitution homomorphism. The domain is the language that provides a term on which the substitution homomorphism is supposed to act. The codomain is the set of functions from syntactic environments to the language.

[4] By the way, in the last equation, we would not object if it turned out that $[y/x]\,(\mu y.x) = \mu x.\, y$, although this is not possible with the classical Curry–Feys definition of substitution [7].

In our specific present setting, our technical results require us to treat, not only the language $L\{X\}$, but also the language consisting of contexts over $L\{X\}$, i.e., elements of $\Omega\{X\}$-algebra $L\{X\}[*]$ of polynomials in one indeterminate $*$ over $L\{X\}$. The details of this approach are somewhat intricate, so we confine ourselves to an outline of the results we need. A complete treatment of these concepts can be found in [15, 17]; while the problems studied there were substitution in the typed and untyped lambda calculi, respectively, the algebraic character of those problems is not essentially different from that in this paper.

The salient points are these:

(i) From Theorem 4.1 of [17], there is an $\Omega\{X\}$-algebra homomorphism

$$\text{Clashset} : L\{X\}[*] \to (L\{X\}[*]^{X\cup\{*\}} \to \mathscr{P}_{\text{fin}}(X \cup \{*\}))$$

that satisfies

$$\text{Clashset}\ C\, e = \bigcup\{\text{Free}^*(e\, y) \mid y \in \text{Free}^*(C)\}.$$

(We remark in passing that the $\Omega\{X\}$-algebra structure of the codomain is not defined pointwise.) The homomorphism Clashset is used to determine when there is a "clash" between the binding operator and the identifiers that should be free in the result of a substitution. If so, the binding operator must be changed.

(ii) Let $e_0 : X \cup \{*\} \to L\{X\}[*]$ be the inclusion map. We can regard

$$G = L\{X\}[*]^{X\cup\{*\}} \to L\{X\}[*],$$

the set of functions from syntactic environments to terms, as an $\Omega\{X\}$-algebra under the following interpretations of the operators in $\Omega\{X\}$:

- $\omega_G \langle g_1, \dots, g_n \rangle\, e = \omega_{L\{X\}[*]} \langle g_1\, e, \dots, g_n\, e \rangle$ for each $\omega \in \Omega$,
- $x_G\, g = g\, x$ for each $x \in X$, and
- $(\mu x_G . g)\, e = \mu z_{L\{X\}[*]} . (g\, [e \mid x \mapsto z_{L\{X\}[*]}])$, where

$$z = \begin{cases} x & \text{if } x \notin S, \\ \text{new}(S) & \text{if } x \in S, \end{cases}$$

$S = \text{Clashset}\, (\mu x_{L\{X\}[*]} . (g\, e_0))\, e$, and new : $\mathscr{P}_{\text{fin}}(X \cup *) \to X \cup \{*\}$ satisfies $\text{new}(S) \notin S$.

(iii) Now, there is an Ω-algebra map $l \mapsto \lambda e.l : L \hookrightarrow G$, and this extends uniquely to a *substitution homomorphism*

$$\mathscr{S} : L\{X\} \to G$$

of $\Omega\{X\}$-algebras, as indicated in the following diagram. (A note of reassurance for the reader: if $p \in L\{X\}$ and $e : X \cup \{*\} \to L\{X\}[*]$, then the value assigned to $*$ by a syntactic environment e simply has no effect on the result $\mathscr{S}\, p\, e$ of performing the substitution, as one would expect from the fact that $*$ does not appear in p.)

(iv) The universal property of $L\{X\}[*]$ as an $\Omega\{X\}$-algebra leads to the definition of the another *substitution homomorphism*

$$\mathcal{S}^* : L\{X\}[*] \to G,$$

which is the unique $\Omega\{X\}$-homomorphism that makes the following diagram commute:

$$\begin{tikzcd}
L\{X\} \arrow[r, "incl"] \arrow[dr, "\mathcal{S}"'] & L\{X\}[*] \arrow[d, "\mathcal{S}^*"'] & \{*\} \arrow[l, "incl"'] \arrow[dl, "* \mapsto \lambda e.e*"] \\
L \arrow[u, "incl"] \arrow[r, "l \mapsto \lambda e.l"'] & G &
\end{tikzcd}$$

The three $\Omega\{X\}$-homomorphisms Clashset, Free* and $\mathcal{S}^*$ are related to one another by the following theorem.

Theorem 4.5 (Oles92a, Theorem 4.10). *For all $C \in L\{X\}[*]$ and all $e \in (L\{X\}[*]^{X\cup\{*\}})$,*

$$\text{Clashset } C\,e = \text{Free}^*(\mathcal{S}^*\, C\, e).$$

Given a context $C \in L\{X\}[*]$ and a term $p \in L\{X\}$, we can define the substitution of p for $*$ in C by

$$C[p] = \mathcal{S}^*\, C\, j_p, \quad \text{where} \quad j_p \in L\{X\}[*]^{X\cup\{*\}} \to L\{X\}[*] \text{ is } j_p = [e_0 \mid * \mapsto p].$$

One indication that this definition works properly is provided by the following result.

Proposition 4.6. *If $p \in L\{X\}$, and $C \in L\{X\}[*]$, then $C[p] \in L\{X\}$.*

Proof. Clearly, $C[p] \in L\{X\}[*]$. Since $* \notin \text{Free}^*\, p$, it follows from Theorem 4.5 that $* \notin \text{Free}^*(C[p])$, and hence that $C[p] \in L\{X\}$. $\square$

A *closed context* is a context over L_c, i.e., a polynomial in the Ω-algebra $L_c[*]$. Closed contexts are the contexts that matter in proving full abstraction results for L_c. Although closed contexts only form an Ω-algebra, they may contain binding operators, so that substituting for $*$ in a closed context makes use of the $\Omega\{X\}$-homomorphism $\mathcal{S}^*$. Thus, if $C \in L_c[*]$ and $p \in L_c$, then

$$C[p] = \mathcal{S}^*\, C\, j_p.$$

Proposition 4.7. (i) $L_c[*] = \{C \in L\{X\}[*] \mid \mathrm{Free}^*(C) = \emptyset\}$.
(ii) *If* $C \in L_c[*]$ *and* $p \in L_c$, *then* $C[p] \in L_c$.

Proof. By using the commutative diagram that defines Free*, it is easy to see that

$$L_c[*] \subseteq \{C \in L\{X\}[*] \mid \mathrm{Free}^*(C) = \emptyset\}.$$

To obtain the reverse inclusion, one proves by induction on n that, for all finite sets S of identifiers,

$$\{C \in L\{X\}[*] \mid \mathrm{Free}^*(C) \subseteq S \text{ and } \mathrm{rank}(C) \leqslant n\} \subseteq L_S.$$

The second part of the proposition then follows from the first part by using Proposition 4.6. □

The following result shows that this approach to substitution and the one presented earlier for Ω-algebras of polynomials over L agree where they overlap.

Theorem 4.8. *Let* $Q \in L[X][*] = L[*][X]$, *so that* Q *can be viewed both as a context over the* Ω*-algebra* $L[X]$ *and as an* Ω*-polynomial in* X *over* $L[*]$.
(i) *If* $x \in X$ *is used to determine via the methods of Section 2 the* Ω*-algebra homomorphism* $[Q/x] : L[*][X] \to L[*][X]$, *then for all* $C \in L[X][*]$,

$$[Q/x]\, C = \mathcal{S}^*\, C\, [e_0 \mid x \mapsto Q].$$

(ii) *Using the inclusion* $\imath : L[X] \to L[X][*]$ *to induce the* Ω*-algebra homomorphism*

$$\imath[* \mapsto Q] : L[X][*] \to L[X][*],$$

we have, for all $C \in L[X][*]$,

$$\imath[* \mapsto Q]\, C = \mathcal{S}^*\, C\, [e_0 \mid * \mapsto Q],$$

thereby showing the notation $C[Q]$ *is unambiguous.*

Proof. Since this result is implicit in [17], we only sketch the proof. For the first part, just show that the function $C \mapsto \mathcal{S}^*\, C\, [e_0 \mid x \mapsto Q]$ from $L[X][*]$ to itself is an Ω-algebra homomorphism that satisfies the universal mapping property that defines $[Q/x]$. For the second part, establish the same relationship between $C \mapsto \mathcal{S}^*\, C\, [e_0 \mid * \mapsto Q]$ and $\imath[* \mapsto Q]$. □

Now, the family $[D \to [D^X \to D]]$ of continuous maps from D to $[D^X \to D]$ is an $\Omega\{X\}$-algebra pointwise. So, again using the universal mapping property for $L\{X\}[*]$, we can define the denotational semantics of $L\{X\}[*]$.

$$\begin{tikzcd}
L\{X\} \arrow[r, "incl"] \arrow[dr, "p \mapsto \lambda d.(\mathscr{D}_{L\{X\}}\, p)"'] & L\{X\}[*] \arrow[d, "\mathscr{S}^*"] & \{*\} \arrow[l, "incl"'] \arrow[dl, "* \mapsto \lambda d.\lambda h.d"] \\
& {[D \to [D^X \to D]]} &
\end{tikzcd}$$

Given a syntactic environment $e \in L\{X\}^{X \cup \{*\}}$ and a semantic environment $h \in D^X$, we define the related semantic environment

$$h_e \in D^X \quad \text{by} \quad h_e x \;=\; \mathscr{D}_{L\{X\}}\,(ex)\,h.$$

The following result plays a crucial role in our development. This theorem sums up the semantic effect of substitution. It expresses the denotational meaning of a context $\mathscr{S}^*\,C\,e$ created by substitution in the presence of a semantic environment h in terms of the meaning of C as well as the meanings of ex as x ranges over $X \cup \{*\}$.

Theorem 4.9. *If $C \in L\{X\}[*]$, $e \in L\{X\}^{X \cup \{*\}}$ and $h \in D^X$, then*

$$\mathscr{D}^*\,C\,(\mathscr{D}_{L\{X\}}\,(e*)\,h)\,h_e = \mathscr{D}_{L\{X\}}\,(\mathscr{S}^*\,C\,e)\,h.$$

Proof. We prove this by rank induction on C. The cases that rank$(C) = 0$ or 1 are routine, as is the case that $C = \omega_{L\{X\}[*]}\langle C_1, \ldots, C_n\rangle$, so we confine our proof to the case that $C = \mu x_{L\{X\}[*]}.C_1$. In this case,

$$\mathscr{D}_{L\{X\}}\,(\mathscr{S}^*\,(\mu x_{L\{X\}[*]}.C_1)\,e)\,h = \mathscr{D}_{L\{X\}}\,(\mu z_{L\{X\}[*]}.(\mathscr{S}^*\,C_1\,[e \mid x \mapsto z]))\,h,$$

where

$$z = \begin{cases} x & \text{if } x \notin S, \\ \mathrm{new}(S) & \text{if } x \in S, \end{cases}$$

and

$$\begin{aligned} S &= \mathrm{Clashset}\,(\mu x_{L\{X\}[*]}.(\mathscr{S}^*\,C_1\,e_0))\,e = \mathrm{Clashset}\,(\mu x_{L\{X\}[*]}.C_1)\,e \\ &= \bigcup\{\mathrm{Free}^*(e\,y) \mid y \in (\mathrm{Free}^*\,C_1) \setminus \{x\}\}. \end{aligned}$$

Hence,

$$\begin{aligned} &\mathscr{D}_{L\{X\}}\,(\mathscr{S}^*\,(\mu x_{L\{X\}[*]}.C_1)\,e)\,h = \mu x_{[D^X \to D]}.(\mathscr{D}_{L\{X\}}\,(\mathscr{S}^*\,C_1\,[e \mid x \mapsto z]))\,h \\ &= \mathrm{Y}_D\,\big(\mathrm{cur}\,((\mathscr{D}_{L\{X\}}\,(\mathscr{S}^*\,C_1\,[e \mid x \mapsto z])) \circ p_z)\;h\big) \\ &= \mathrm{Y}_D\,(d \mapsto \mathscr{D}_{L\{X\}}\,(\mathscr{S}^*\,C_1\,[e \mid x \mapsto z])\,[h \mid z \mapsto d]). \end{aligned}$$

On the other hand,

$$\begin{aligned}
&\mathscr{D}^*(\mu x_{L\{X\}[*]}.C_1)(\mathscr{D}_{L\{X\}}(e*)h)h_e\\
&= \mu x_{[D\to[D^X\to D]]}.(\mathscr{D}^*C_1)(\mathscr{D}_{L\{X\}}(e*)h)h_e\\
&= \mu x_{[D^X\to D]}.(\mathscr{D}^*C_1(\mathscr{D}_{L\{X\}}(e*)h))h_e\\
&= \mathrm{Y}_D\left(\mathrm{cur}((\mathscr{D}^*C_1(\mathscr{D}_{L\{X\}}(e*)h))\circ p_x)\,h_e\right)\\
&= \mathrm{Y}_D(d\mapsto \mathscr{D}^*C_1(\mathscr{D}_{L\{X\}}(e*)h)[h_e\mid x\mapsto d]).
\end{aligned}$$

Now,

$$\begin{aligned}
[h_e\mid x\mapsto d]x = d = \mathscr{D}_{L\{X\}}z[h\mid z\mapsto d] &= \mathscr{D}_{L\{X\}}([e\mid x\mapsto z]x)[h\mid z\mapsto d]\\
&= [h\mid z\mapsto d]_{[e\mid x\mapsto z]}x,
\end{aligned}$$

while, for $y\neq x$,

$$[h_e\mid x\mapsto d]y = \mathscr{D}_{L\{X\}}(e\,y)h = \mathscr{D}_{L\{X\}}(e\,y)[h\mid z\mapsto d]$$

since we know $z\notin \mathrm{Free}^*(e\,y)$. Thus, we conclude

$$\begin{aligned}
[h_e\mid x\mapsto d]y = \mathscr{D}_{L\{X\}}(ey)[h\mid z\mapsto d] &= \mathscr{D}_{L\{X\}}([e\mid x\mapsto z]y)[h\mid z\mapsto d]\\
&= [h\mid z\mapsto d]_{[e\mid x\mapsto z]}y.
\end{aligned}$$

Since, $e* = [e\mid x\mapsto z]*$, for each $d\in D$, we can conclude that

$$\mathscr{D}^*C_1(\mathscr{D}_{L\{X\}}(e*)h)[h_e\mid x\mapsto d] = \mathscr{D}^*C_1(\mathscr{D}_{L\{X\}}([e\mid x\mapsto z]*)h)[h_e\mid x\mapsto d],$$

so the inductive hypothesis and our calculation above imply that

$$\mathscr{D}^*C_1(\mathscr{D}_{L\{X\}}(e*)h)[h_e\mid x\mapsto d] = \mathscr{D}_{L\{X\}}(\mathscr{S}^*C_1[e\mid x\mapsto z])[h\mid z\mapsto d],$$

and this means that

$$\begin{aligned}
&\mathscr{D}_{L\{X\}}(\mathscr{S}^*(\mu x_{L\{X\}[*]}C_1).e)h\\
&\quad= \mathrm{Y}_D(d\mapsto \mathscr{D}_{L\{X\}}(\mathscr{S}^*C_1[e\mid x\mapsto z])[h\mid z\mapsto d])\\
&\quad= \mathrm{Y}_D(d\mapsto \mathscr{D}^*C_1(\mathscr{D}_{L\{X\}}(e*)h)[h_e\mid x\mapsto d])\\
&\quad= \mathscr{D}^*(\mu x_{L\{X\}[*]}C_1).(\mathscr{D}_{L\{X\}}(e*)h)h_e.
\end{aligned}$$

This proves the result. □

The following corollary generalizes Lemma 7.14(b) of [3].

Corollary 4.10. *If* $p\in L\{X\}$, $e\in L\{X\}^{X\cup\{*\}}$, *and* $h\in D^X$, *then*

$$\mathscr{D}_{L\{X\}}\,p\,h_e = \mathscr{D}_{L\{X\}}(\mathscr{S}\,p\,e)h.$$

Proof.

$$\begin{aligned}\mathscr{D}_{L\{X\}}\, p\, h_e &= (\lambda d.(\mathscr{D}_{L\{X\}}\, p))(\mathscr{D}_{L\{X\}}\,(e*)\,h)\,h_e\\ &= \mathscr{D}^*\, p\,(\mathscr{D}_{L\{X\}}\,(e*)\,h)\,h_e\\ &= \mathscr{D}_{L\{X\}}\,(\mathscr{S}^*\, p\, e)\,h\\ &= \mathscr{D}_{L\{X\}}\,(\mathscr{S}\, p\, e)\,h. \qquad \square\end{aligned}$$

Corollary 4.11. *If $C \in L\{X\}[*]$, $p \in L\{X\}$ and $h \in D^X$, then*

$$\mathscr{D}^*\, C\,(\mathscr{D}_{L\{X\}}\, p\, h)\,h = \mathscr{D}_{L\{X\}}\,(C[p])\,h.$$

Proof. If we take $e = j_p$ and let $h \in D^X$, then we note that

$$h_{j_p} x = \mathscr{D}_{L\{X\}}\,(j_p x)\,h = hx,$$

since $j_p x = x$ for every $x \in X$. Taking $e = j_p$ in the preceding theorem, we find that

$$\begin{aligned}\mathscr{D}^*\, C\,(\mathscr{D}_{L\{X\}}\, p\, h)\,h &= \mathscr{D}^*\, C\,(\mathscr{D}_{L\{X\}}\,(j_p *)\,h)\,h_{j_p}\\ &= \mathscr{D}_{L\{X\}}\,(\mathscr{S}^*\, C\, j_p)\,h = \mathscr{D}_{L\{X\}}\,(C[p])\,h. \qquad \square\end{aligned}$$

Our final results in this section bring the mappings $\pi_n : L\{X\} \to L$ into the picture. When we introduced these mappings in Section 2, we indicated that they played a role in relating the operational meaning of a recursive term to its denotational meaning. And, in the next section, we use the π_n's to extend the operational semantics $\mathscr{O} : L \to O$ to an operational semantics for L_c. The following results provide the link we need to guarantee that the extended operational semantics is consistent with the denotational semantics we have derived.

Lemma 4.12. *If $p, q \in L[X]$ and $h \in D^X$, then $\mathscr{D}_{L[X]}([p/x]q)\,h = \mathscr{D}_{L[X]}\, q\,[h \mid x \mapsto \mathscr{D}_{L[X]}\, p\, h]$ provided $x \notin \mathrm{Free}(p)$.*

Proof. To be perfectly clear about it, in this lemma $[p/x]$ is an Ω-algebra homomorphism from $L[*][X]$ to itself.

We want to use Corollary 4.11. Let $p, q \in L[X]$ and $h \in D^X$ be given. We first use the Ω-algebra homomorphism $[*/x] : L[*][X] \to L[*][X]$ to define the context

$$Q \in L[X][*] \quad \text{by} \quad Q = [*/x]\, q$$

Just as in the proof of Proposition 2.20, we see that $q = Q[x]$ and $x \notin \mathrm{Free}^*(Q)$.

Claim. $[p/x]q = Q[p]$.

Indeed, this can be shown by a simple computation, where $\imath : L[X] \to L[X][*]$ is the inclusion:

$$\begin{aligned}[p/x]\,q &= [p/x]\,(Q[x]) \\ &= [p/x]\,(\imath[* \mapsto x]\,Q) \\ &= ([p/x] \circ \imath)[* \mapsto p]\,Q \\ &= \imath[* \mapsto p]\,Q \\ &= Q[p],\end{aligned}$$

where the sequence of equalities have the following justifications:

(i) By the construction of Q.
(ii) By the definition of $Q[x]$.
(iii) By Corollary 2.12.
(iv) By Lemma 2.19, which applies because $([p/x] \circ \imath)|_{\mathrm{Free}^*(Q)} = \imath|_{\mathrm{Free}^*(Q)}$.
(v) By the definition of $Q[p]$.

Now, to prove this lemma, we note that, for $h \in D^X$,

$$\mathscr{D}_{L[X]}([p/x]q)\,h = \mathscr{D}_{L\{X\}}\,(Q[p])\,h = \mathscr{D}^*\,Q\,(\mathscr{D}_{L[X]}\,p\,h)\,h,$$

the last equality following from Corollary 4.11.

Likewise,

$$\begin{aligned}&\mathscr{D}_{L[X]}\,q\,[h \mid x \mapsto \mathscr{D}_{L[X]}\,p\,h] \\ &\quad = \mathscr{D}_{L\{X\}}\,q\,[h \mid x \mapsto \mathscr{D}_{L[X]}\,p\,h] \\ &\quad = \mathscr{D}_{L[X]}\,(Q[x])\,[h \mid x \mapsto \mathscr{D}_{L[X]}\,p\,h] \\ &\quad = \mathscr{D}^*\,Q\,(\mathscr{D}_{L[X]}x\,[h \mid x \mapsto \mathscr{D}_{L[X]}\,p\,h])\,[h \mid x \mapsto \mathscr{D}_{L[X]}\,p\,h],\end{aligned}$$

the last equality again following from Corollary 4.11. Moreover, since $x \notin \mathrm{Free}^*(Q)$,

$$\begin{aligned}&\mathscr{D}_{L[X]}\,q\,[h \mid x \mapsto \mathscr{D}_{L[X]}\,p\,h] \\ &\quad = \mathscr{D}^*\,Q\,(\mathscr{D}_{L[X]}x\,[h \mid x \mapsto \mathscr{D}_{L[X]}\,p\,h])\,[h \mid x \mapsto \mathscr{D}_{L[X]}\,p\,h] \\ &\quad = \mathscr{D}^*\,Q\,(\mathscr{D}_{L[X]}\,p\,h)\,h,\end{aligned}$$

so that both sides of the desired equation are equal to the same thing, namely, $\mathscr{D}^*\,Q\,(\mathscr{D}_{L[X]}\,p\,h)\,h$. This proves the lemma. □

Theorem 4.13. *The mapping* $\mathscr{D}_{L\{X\}} : L\{X\} \to [D^X \to D]$ *satisfies*

$$\mathscr{D}_{L\{X\}} = \bigsqcup_{n \in \mathbb{N}} \mathscr{D}_{L[X]} \circ \pi_n.$$

Proof. Fix $p \in L\{X\}$. We prove the result by induction on the rank of p. The cases that $\mathrm{rank}(p) \leqslant 1$ are easy, since all π_n's are the identity map on these terms. And, the case that $p = \omega_{L\{X\}}\langle p_1, \ldots, p_m\rangle$ also follows routinely. So, we consider only the case that $p = \mu x_{L\{X\}}.\,p'$, knowing that the result holds for p'.

Fix $h \in D^X$, and let $d = \mathscr{D}_{L\{X\}}\,(\mu x_{L\{X\}}.\,p')\,h \in D$. Since $\mathscr{D}_{L\{X\}}$ is a homomorphism, $d = \mu x_{[D^X \to D]}.(\mathscr{D}_{L\{X\}}\,p')\,h$, and so d is the least fixed point of the function $F(d') = \mathscr{D}_{L\{X\}}\,p'\,[h \mid x \mapsto d']$. Hence, $d = F(d) = \bigsqcup_{n \in \mathbb{N}} d_n$, where $d_0 = \perp_D$ and $d_{n+1} = F(d_n)$.

Now, let $\Delta = \{\delta_{m,n} \mid m \leqslant n \in \mathbb{N}\}$ be defined recursively by

- $\delta_{0,n} = \perp_D$ for all $n \in \mathbb{N}$, and
- $\delta_{m+1,n+1} = \mathscr{D}_{L\{X\}}\,(\pi_n(p'))\,[h \mid x \mapsto \delta_{m,n}]$ for $m, n > 0$.

The fact that the π_n's are monotone with respect to the minimum monotone pre-order on $L\{X\}$, together with the fact that $\mathscr{D}_{L\{X\}}$ is a homomorphism imply that Δ is directed by the product order on $\mathbb{N} \times \mathbb{N}$ (i.e., $\langle m,n\rangle \leqslant \langle m',n'\rangle$ iff $m \leqslant m'$ and $n \leqslant n'$).

Claim 1. For each $n \in \mathbb{N}$, $d_m = \bigsqcup_{n \geqslant m} \delta_{m,n}$.

We prove this by induction on $m \in \mathbb{N}$. For the case $m = 0$, the result is clear. Assume it holds for some m, and consider d_{m+1}. By definition,

$$d_{m+1} = \mathscr{D}_{L\{X\}}\,p'\,[h \mid x \mapsto d_m] = \left(\bigsqcup_{n \in \mathbb{N}} \mathscr{D}_{L[X]}\,(\pi_n(p'))\right)\left[h \mid x \mapsto \bigsqcup_{k \geqslant m} \delta_{m,k}\right],$$

the last equality following from the inductive hypotheses on p and m. Since all the mappings and operations are continuous, we can bring both suprema in the last term to the outside, obtaining,

$$d_{m+1} = \bigsqcup_{n \in \mathbb{N}} \bigsqcup_{k \geqslant m} \mathscr{D}_{L[X]}\,(\pi_n(p'))\,[h \mid x \mapsto \delta_{m,k}].$$

We claim that this last term is equal to $\bigsqcup_{n \geqslant m+1} \delta_{m+1,n}$. Indeed, each term

$$\delta_{m+1,n} = \mathscr{D}_{L\{X\}}\,(\pi_{n-1}(p'))\,[h \mid x \mapsto \delta_{m,n-1}]$$

occurs in the family of terms

$$Y = \{\mathscr{D}_{L[X]}\,(\pi_n(p'))\,[h \mid x \mapsto \delta_{m,k}] \mid n \in \mathbb{N}, k \geqslant m\}$$

so the supremum of the $\delta_{m+1,n}$'s is at most the value of d_{m+1}. But, conversely, each term in Y is dominated by some term $\delta_{m+1,n}$ for some $n \geqslant m+1$, since Δ is directed. Hence

$$d_{m+1} = \bigsqcup_{n \in \mathbb{N}} \bigsqcup_{k \geqslant m} \mathscr{D}_{L[X]}\,(\pi_n(p'))\,[h \mid x \mapsto \delta_{m,k}] = \bigsqcup_{n \geqslant m+1} \delta_{m+1,n},$$

which proves the inductive step, and hence proves the claim.

Claim 2. For each $n \in \mathbb{N}$, $\delta_{n,n} = \mathscr{D}_{L[X]}\,(\pi_n(\mu x.p'))\,h$.

We prove this by induction on $n \in \mathbb{N}$. For $n = 0$, the result is trivial. Assume it holds for some $n \in \mathbb{N}$. Then,

$$\begin{aligned}\delta_{n+1,n+1} &= \mathscr{D}_{L[X]}\,(\pi_n(p'))\,[h \mid x \mapsto \delta_{n,n}]\\ &= \mathscr{D}_{L[X]}\,(\pi_n(p'))\,[h \mid x \mapsto \mathscr{D}_{L[X]}\,(\pi_n(\mu x.p'))\,h]\end{aligned}$$

$$= \mathscr{D}_{L[X]}([\pi_n(\mu x.p')/x](\pi_n(p')))\,h$$
$$= \mathscr{D}_{L[X]}(\pi_{n+1}(\mu x.p'))\,h,$$

the last equality following from the definition of π_{n+1}.

The inductive step now follows, since

$$\begin{aligned}\mathscr{D}_{L\{X\}}\ \mu x_{L\{X\}}.\ p'\ h = d &= \bigsqcup_m d_m \\ &= \bigsqcup_m \bigsqcup_{n\geqslant m+1} \delta_{m+1,n} \\ &= \bigsqcup_m \mathscr{D}_{L[X]}(\pi_m(\mu x.p'))\,h.\end{aligned}$$

The result then follows by induction on rank(p). $\square$

Corollary 4.14. *If $p \in L_c$, then $\mathscr{D}_c(p) = \bigsqcup_{n\in\mathbb{N}} \mathscr{D}(\pi_n(p))$.*

Example 4.15 (*continued*). Let us again consider our example language S. Corollary 4.14 makes it easy to compute the denotational meanings of simple programs involving recursion. Let a be an atomic action.

(i) Let $p_1 = \mu x.x$. Then, for all n, $\pi_0(p_1) = b_S$. Hence $\mathscr{D}_c(p_1) = \perp_D$.

(ii) Let $p_2 = \mu x.(a\,;x)$. Then

$$\pi_0(p_2) = b_S,\ \ \pi_1(p_2) = a\,;b_S,\ \ \pi_2(p_2) = a\,;a\,;b_S,\ \ \pi_3(p_2) = a\,;a\,;a\,;b_S, \ldots,$$

so that

$$\mathscr{D}_c(p_2) = \bigcup\{[\{a^n\}] \mid n \in \mathbb{N}\}.$$

(iii) Let $p_3 = \mu x.(\delta_S + (x\,;a))$. Using that δ_S is a zero for $+$,

$$\pi_0(p_3) = b_S,\ \ \pi_1(p_3) = b_S\,;a,\ \ \pi_2(p_3) = b_S\,;a\,;a,\ \ \pi_3(p_3) = b_S\,;a\,;a\,;a, \ldots,$$

so that

$$\mathscr{D}_c(p_3) = \perp_D \neq \mathscr{D}_c(\delta_S).$$

(iv) Let $p_4 = \mu x.(\varepsilon_S + (x\,;a))$. Using right distributivity of $+$ over ;, which holds in S, we see

$$\pi_0(p_3) = b_S, \quad \pi_1(p_4) = \varepsilon_S + (b_S\,;a), \quad \pi_2(p_4) = \varepsilon_S + a + (b_S\,;a\,;a),$$

$$\pi_3(p_4) = \varepsilon + a + (a\,;a) + (b_S\,;a\,;a\,;a), \ldots,$$

so that

$$\mathscr{D}_c(p_4) = \bigcup\{[\{a^n\surd\}] \mid n \in \mathbb{N}\}.$$

In particular, $\mathscr{D}_c(p_2)$ is a proper subset of $\mathscr{D}_c(p_4)$.

5. Operational semantics of L_c

In the last section we gave a denotational semantics for the language L_c of closed terms in $L\{X\}$ based on some simple assumptions about the denotational model of L. We now show how to extend the operational model $\mathcal{O} : L \to O$ to one for $L_c$using the following assumptions that pertain only to the operational model of L: the mapping $\mathcal{O} : L \to O$ is an ordered operational model and O is a cpo.

In order to extend $\mathcal{O}$ to L_c, it is necessary to define the behavior of recursive terms, $\mu x.p$ (as well as the other terms of L_c). This is where the mappings π_n enter the picture. Proposition 2.18 shows that each term of L_c has its image in L under π_n for each $n \in \mathbb{N}$. And, Proposition 2.16 implies that the sequence $\{\pi_n(p)\}_{n\in\mathbb{N}}$ is increasing with respect to $\preceq_L$ for each term $p \in L_c$. So, the monotonicity property of $\mathcal{O}$ means that the sequence $\{\mathcal{O}(\pi_n(p))\}_{n\in\mathbb{N}}$ is increasing in O. Hence, we can use this sequence to define the behavior of $\mathcal{O}_c(p)$ for each $p \in L_c$.

Definition 5.1. We define the mapping $\mathcal{O}_c : L_c \to O$ by $\mathcal{O}_c(p) = \bigsqcup_n \mathcal{O}(\pi_n(p))$ for each $p \in L_c$.

Proposition 5.2. *The mapping $\mathcal{O}_c : L_c \to O$ is monotone with respect to $\preceq_{L_c}$, and satisfies $\mathcal{O}_c(p) = \mathcal{O}(p)$ for each $p \in L$. Hence $\mathcal{O}_c : L_c \to O$ is an ordered operational model for L_c.*

Proof. Recall that L_c is an Ω-algebra by definition, and so Corollary 2.4 shows that, if $p \preceq_{L_c} q$, then $\pi_n(p) \preceq_L \pi_n(q)$ for each $n \in \mathbb{N}$. Thus,

$$\mathcal{O}_c(p) = \bigsqcup_{n\in\mathbb{N}} \mathcal{O}(\pi_n(p)) \leqslant_O \bigsqcup_{n\in\mathbb{N}} \mathcal{O}(\pi_n(q)) = \mathcal{O}_c(q).$$

Also, if $p \in L$, then the definition of π_n implies $\pi_n|_L = \mathbf{1}_L$, and so

$$\mathcal{O}_c(p) = \bigsqcup_{n\in\mathbb{N}} \mathcal{O}(\pi_n(p)) = \bigsqcup_{n\in\mathbb{N}} \mathcal{O}(p) = \mathcal{O}(p). \quad \square$$

Corollary 4.4 and Proposition 5.2 show that mild assumptions about D and O that one would expect when both are cpo's allow us to give both a denotational model and a clearly defined operational meaning for all closed terms in L_c.

Example 5.3 (*continued*). Returning to our example language S and its extension S_c, we see it is clear from Theorem 3.8, Corollary 4.14, and the definition of O_c that $O_c = \mathscr{D}_c$. In particular, $\mathscr{D}_c$ is obviously fully abstract with respect to O_c. Although full abstraction is not a serious issue for S_c, it is the use of the algebraic approach developed here that makes this observation so easy to prove. Basically, all the work was done in studying the semantics of the ground language S.

6. Order adequacy and strong order full abstraction

In this section we assume the conditions needed for the last two sections:

(i) $\mathscr{D} : L \to D$ is an ordered denotational model for L and D is a continuous Ω-algebra.

(ii) $\mathscr{O} : L \to O$ is an ordered operational model for L and O is a cpo.

Having set the stage by extending the operational model $\mathscr{O} : L \to O$ to an operational model $\mathscr{O}_c : L_c \to O$, and extending the denotational model $\mathscr{D} : L \to D$ to a denotational model $\mathscr{D}_c : L_c \to D$, we now are ready to establish the main results of the paper. In this section, we first explain when the order adequacy relation for models of L to can be lifted to of L_c, and then we do the same for strong order full abstraction. Crucial to our arguments are two conditions mentioned in the Introduction. Recall the *denotational finiteness condition:*

- for all $p \in L$ and $q \in L_c$, $\mathscr{D}(p) \leqslant \bigsqcup\{\mathscr{D}(\pi_n(q)) | n \in \mathbb{N}\}$ implies that there exists some $m \in \mathbb{N}$ such that $\mathscr{D}(p) \leqslant \mathscr{D}(\pi_m(q))$

and the similar *operational finiteness condition:*

- for all $p \in L$ and $q \in L_c$, $\mathscr{O}(p) \leqslant \bigsqcup\{\mathscr{O}(\pi_n(q)) | n \in \mathbb{N}\}$ implies that there exists some $m \in \mathbb{N}$ such that $\mathscr{O}(p) \leqslant \mathscr{O}(\pi_m(q))$.

First we prove that $\mathscr{D}_c$ is order adequate with respect to $\mathscr{O}_c$.

Theorem 6.1. *If $\mathscr{D}$ is order adequate with respect to $\mathscr{O}$ and the denotational finiteness condition holds, then $\mathscr{D}_c$ is order adequate with respect to $\mathscr{O}_c$.*

Proof. Suppose that $p, q \in L_c$ with $\mathscr{D}_c(p) \leqslant_D \mathscr{D}_c(q)$, and let $n \in \mathbb{N}$. By Corollary 4.14,

$$\mathscr{D}(\pi_n(p)) \leqslant_D \bigsqcup_{m \in \mathbb{N}} \mathscr{D}(\pi_m(p)) = \mathscr{D}(p) \leqslant_D \mathscr{D}(q) = \bigsqcup_{m \in \mathbb{N}} \mathscr{D}(\pi_m(q)).$$

Hence, there exists an integer $k(n)$ such that, for all integers $m \geqslant k(n)$, $\mathscr{D}(\pi_n(p)) \leqslant_D \mathscr{D}(\pi_m(q))$. By order adequacy of $\mathscr{D}$ with respect to $\mathscr{O}$, for all integers $m \geqslant k(n)$, $\mathscr{O}(\pi_n(p)) \leqslant_O \mathscr{O}(\pi_m(q))$. Hence, $\mathscr{O}(\pi_n(p)) \leqslant_O \bigsqcup_{m \in \mathbb{N}} \mathscr{O}(\pi_m(q)) = \mathscr{O}_c(q)$.

Thus,

$$\mathscr{O}_c(p) = \bigsqcup_{n \in \mathbb{N}} \mathscr{O}(\pi_n(p)) \leqslant_O \mathscr{O}_c(q),$$

which proves the desired result. □

Closed contexts enter into the statement of the following strengthened version of order adequacy, which is the converse of order full abstraction.

Proposition 6.2. *Assume $\mathscr{D}$ is order adequate with respect to $\mathscr{O}$ and the denotational finiteness condition holds. Let $p, q \in L_c$. If $\mathscr{D}_c(p) \leqslant_D \mathscr{D}_c(q)$, then, for all closed contexts $C \in L_c[*]$, $\mathscr{O}_c(C[p]) \leqslant_O \mathscr{O}_c(C[q])$.*

Proof. Suppose $\mathscr{D}_c(p) \leqslant_D \mathscr{D}_c(q)$. Let C be a closed context and let $h \in D^X$. We then have

$$\begin{aligned}
\mathscr{D}_c(C[p]) &= (\kappa_D \circ \mathscr{D}_c)(C[p])h \\
&= \mathscr{D}_{L\{X\}}\ (C[p])h \\
&= \mathscr{D}^*\,C\,(\mathscr{D}_{L\{X\}}\ p\,h)h \\
&= \mathscr{D}^*\,C\,(\mathscr{D}_c\,p)h \\
&\leqslant_D \mathscr{D}^*\,C\,(\mathscr{D}_c\,q)h \\
&= \cdots \\
&= \mathscr{D}_c(C[p]),
\end{aligned}$$

where the justifications for the steps of the computation up to the $\cdots$ are as follows:

(i) By the definition of κ_D.
(ii) By Corollary 4.4.
(iii) By Corollary 4.11.
(iv) By Corollary 4.4 and the definition of κ_D.
(v) Since $\mathscr{D}^*C$ is continuous, and, hence, monotone.

Finally, order adequacy of $\mathscr{D}_c$ with respect to $\mathscr{O}_c$ gives $\mathscr{O}_c(C[p]) \leqslant_O \mathscr{O}_c(C[q])$. $\square$

The next theorem shows that strong order full abstraction can be lifted from $\mathscr{D}$ and $\mathscr{O}$ to $\mathscr{D}_c$ and $\mathscr{O}_c$. An important point that ensures that the proof goes through is that the constructed closed context C actually turns out to be in $L[*]$.

Theorem 6.3. *Assume $\mathscr{D}$ is strongly order fully abstract with respect to $\mathscr{O}$ and the operational finiteness condition holds.*

(i) *If $p, q \in L_c$ and $\mathscr{D}_c(p) \not\leqslant_D \mathscr{D}_c(q)$, then there is some context $C \in L[*]$ satisfying $\mathscr{O}_c(C[p]) \not\leqslant_O \mathscr{O}_c(C[q'])$ for every $q' \in L_c$ with $q \preceq_{L_c} q'$.*

(ii) *$\mathscr{D}_c$ is strongly order fully abstract with respect to $\mathscr{O}_c$.*

Proof. Assume that $\mathscr{D}_c(p) \not\leqslant_D \mathscr{D}_c(q)$ for $p, q \in L_c$. Since we have $\mathscr{D}_c(p) = \bigsqcup_{n\in\mathbb{N}} \mathscr{D}(\pi_n(p))$, it follows that there is some $m \in \mathbb{N}$ such that $\mathscr{D}(\pi_m(p)) \not\leqslant_D \mathscr{D}_c(q)$. Then, $\mathscr{D}_c(q) = \bigsqcup_{n\in\mathbb{N}} \mathscr{D}(\pi_n(q))$ implies that $\mathscr{D}(\pi_m(p)) \not\leqslant_D \mathscr{D}(\pi_m(q))$. We now apply the fact that $\mathscr{D}$ is strongly order fully abstract with respect to $\mathscr{O}$ to conclude there is some context $C \in L[*]$ with $\mathscr{O}(C[\pi_m(p)]) \not\leqslant_O \mathscr{O}(C[q'])$ for any $q' \in L$ with $\pi_m(q) \preceq_L q'$. If $q' \in L_c$ satisfies $q \preceq_{L_c} q'$, then $\pi_n(q) \preceq_L \pi_n(q')$ for each $n \in \mathbb{N}$. Hence, $\pi_m(q) \preceq_L \pi_{m+n}(q')$ for each $n \in \mathbb{N}$, so $\mathscr{O}(C[\pi_m(p)]) \not\leqslant_O \mathscr{O}(C[\pi_{m+n}(q')])$ for each $n \in \mathbb{N}$. We apply Proposition 2.15 to conclude that $\mathscr{O}(C[\pi_m(p)]) = \mathscr{O}(\pi_m(C[p]))$ and $\mathscr{O}(C[\pi_{m+n}(q')]) = \mathscr{O}(\pi_{m+n}(C[q']))$ for each $n \in \mathbb{N}$. By the operational finiteness condition, $\mathscr{O}(\pi_m(C[p])) \not\leqslant_O \bigsqcup_{n\in\mathbb{N}} \mathscr{O}(C[\pi_{m+n}(q')]) = \mathscr{O}_c(C[q'])$. Since $\mathscr{O}(\pi_m(C[p])) \leqslant_O \mathscr{O}_c(C[p])$, it follows that $\mathscr{O}_c(C[p]) \not\leqslant_O \mathscr{O}_c(C[q'])$.

Since $C \in L[*] \subseteq L\{X\}[*]$, C has no occurrences of any identifiers at all, and this means $C \in L_c[*]$. From this, strong order full abstraction of $\mathscr{D}_c$ with respect to $\mathscr{O}_c$ follows. $\square$

7. The extension to algebras satisfying equations

The principal results so far are Theorems 6.1 and 6.3, which give conditions under which order adequacy and strong order full abstraction results for a ground language L can be lifted to corresponding results for an extension language L_c that supports recursion. However, the language L_c of closed terms is anomic, so that any equational laws valid in L do not hold in L_c. In this section, we extend Theorems 6.1 and 6.3 to languages satisfying equations.

In this section, we suppose L is a language that satisfies a family of equational laws, E, so that L is an (Ω, E)-algebra. We also assume that we are given an ordered operational model $\mathcal{O} : L \to O$, where O is a cpo, and a related ordered denotational model $\mathcal{D} : L \to D$, where D is a cpo that is a continuous (Ω, E)-algebra.

Since all the operators that appear in the equations in E are in $\Omega\{X\}$ as well as Ω, we also can consider $(\Omega\{X\}, E)$-algebras when the need arises. Given an algebra A, we will denote by A_E the algebra of congruence classes with respect to the smallest congruence generated by E. In particular, if A is an Ω-algebra, then A_E is the Ω-algebra quotient of A, while if A is an $\Omega\{X\}$-algebra, then A has an $\Omega\{X\}$-algebra structure. By $\Phi_A : A \to A_E$ we mean the projection homomorphism in the appropriate category of algebras.

None of the extensions $L[X]$, $L\{X\}$, and L_c that we have studied satisfy the set E of equations satisfied by L. From a programming language viewpoint, it might seem preferable to assume that any laws that hold in the base language L also hold in extensions supporting the addition of identifiers and recursion. Thus, we would like to consider the semantics of L_{cE}, which, according to our notation, is the $\Omega\{X\}$-algebra L_c of closed terms modulo (the congruence generated by) E.

Before we go further, we should explain why we first have shown our results for algebras without equational laws. We could say that it simply was easier not to lug a set of equations through the proofs, and this surely would be true. However there is an important technical reason why this is not possible, and it is connected to our heavy reliance on algebraic techniques. We have made extensive use of rank functions, i.e., homomorphisms into $\mathbb{N}$. If the domain of a rank function satisfies some arbitrary set of equations, then its image, which we expect to be an infinite subset of $\mathbb{N}$, also would have to satisfy the same set of equations. But, we simply do not see how to define usable rank functions in such generality. In particular, while we can define Ω-algebra structures on $\mathbb{N}$ when necessary, we do not see how to make $\mathbb{N}$ into an (Ω, E)-algebra. So we are left with the course actually taken: to prove the desired results in the anomic case, and then use universal algebra to extend them to the equational case.

The extensions to L we study in this section are

(i) $L[X]_E$, the Ω-algebra $L[X]$ modulo E,

(ii) $L\{X\}_E$, the $\Omega\{X\}$-algebra $L\{X\}$ modulo E, and

(iii) L_{cE}, the $\Omega\{X\}$-algebra L_c modulo E.

Are these quotient algebras truly extensions of L? Since each of $L[X]$, $L\{X\}$, and L_c contains L as an Ω-subalgebra, and L satisfies E, no elements of L are identified on passing to the quotient algebras, and, thus, we may assume L is an Ω-subalgebra of each quotient.

Since $L\{X\}$ and L_c are anomic algebras, the congruence on L_c generated by E is the restriction to L_c of the congruence on $L\{X\}$ generated by E. This enables us to apply the Second Isomorphism Theorem ([6, p.8]) to conclude that L_{cE} may be identified with $\Phi_{L\{X\}}(L_c)$, the image in $L\{X\}_E$ of L_c under the projection homomorphism. A similar argument shows us that $L[X]_E$ may be identified with $\Phi_{L\{X\}}(L[X])$, the projection of $L[X]$ into $L\{X\}_E$.

The (Ω, E)-algebra $L[X]_E$ can be regarded as a polynomial algebra like the Ω-algebra $L[X]$ because it satisfies a similar universal mapping property arising from the fact that $L[X]_E$ is the coproduct of L and the free algebra $T_{\Omega,E}[X]$ generated by X in the category of (Ω, E)-algebras. In like fashion, $L\{X\}_E$ satisfies a universal mapping property similar to that satisfied by $L\{X\}$ because $L\{X\}_E$ is the free $(\Omega\{X\}, E)$-algebra generated by the (Ω, E)-algebra L. Finally, the fact that $L\{X\}$, when regarded only as an Ω-algebra, is a polynomial algebra over L generated by $X \cup \{\mu x.p \mid x \in X,\ p \in L\{X\}\}$ generalizes to $L\{X\}_E$: in the category of (Ω, E)-algebras, $L\{X\}_E$ is a polynomial algebra over L generated by $X \cup \{\mu x.p \mid x \in X,\ p \in L\{X\}\}$.

Using the same reasoning as in the definition of the mappings π_n in Section 2, we can define (Ω, E)-algebra homomorphisms $\psi_n : L\{X\}_E \to L[X]_E$ as follows:

- $\psi_0 : L\{X\}_E \to L[X]_E$ is the homomorphism determined by
  - (i) the inclusion $f = \iota : L \hookrightarrow L[X]_E$,
  - (ii) the inclusion map $x \mapsto x : X \hookrightarrow L[X]_E$, and
  - (iii) the constant map $\mu x.p \mapsto b_L$ for each $x \in X$ and each $p \in L\{X\}_E$.
- $\psi_{n+1} : L\{X\}_E \to L[X]_E$ is the homomorphism determined by
  - (i) the inclusion $f = \iota : L \hookrightarrow L[X]_E$,
  - (ii) the inclusion map $x \mapsto x : X \hookrightarrow L[X]_E$, and
  - (iii) the mapping $\mu x.p \mapsto [\psi_n(\mu x.p)/x](\psi_n(p))$ for each $x \in X$ and each $p \in L\{X\}_E$.

As in the case of the π_n's, the well-definedness of $\psi_{n+1}(p)$ depends on the fact that $\psi_n(\mu x.p)$ actually is in $L[X]_E$, which ensures that the substitution $[\psi_n(\mu x.p)/x]$ is a self-map of $L[X]_E$.

Lemma 7.1. *For each $n \in \mathbb{N}$, $\psi_n \circ \Phi_{L\{X\}} = \Phi_{L[X]} \circ \pi_n$.*

Proof. We proceed by induction on $n \in \mathbb{N}$. To begin, $\Phi_{L[X]} \circ \pi_0 : L\{X\} \to L[X]_E$ is an Ω-algebra mapping into the (Ω, E)-algebra $L[X]_E$. It is routine to show that

- $\Phi_{L[X]} \circ \pi_0 \circ \iota : L \to L[X]_E$ is the natural inclusion in the category of (Ω, E)-algebras,
- $\Phi_{L[X]}(\pi_0(x)) = x : X \to L[X]_E$ is the natural inclusion, and
- $\Phi_{L[X]}(\pi_0(\mu x.p)) = b_L$ for each $x \in X$ and each $p \in L\{X\}$.

Since $L[X]_E$ is an (Ω, E)-algebra, this implies that there is an (Ω, E)-algebra map $\psi'_0 : L\{X\}_E \to L[X]_E$ satisfying $\psi'_0 \circ \Phi_{L\{X\}} = \Phi_{L[X]} \circ \pi_0$. The conditions just listed for $\Phi_{L[X]} \circ \pi_0$ imply that ψ'_0 satisfies the defining conditions listed above for ψ_0, and since these conditions uniquely determine ψ_0, the mappings are equal. Thus $\psi_0 \circ \Phi_{L\{X\}} = \psi'_0 \circ \Phi_{L\{X\}} = \Phi_{L[X]} \circ \pi_0$.

For the inductive step, assume it has been shown that $\psi_n \circ \Phi_{L\{X\}} = \Phi_{L[X]} \circ \pi_n$. Again, it is routine to show that

- $\Phi_{L[X]} \circ \pi_{n+1} \circ \iota : L \to L[X]_E$ is the natural inclusion in the category of (Ω, E)-algebras,
- $\Phi_{L[X]}(\pi_{n+1}(x)) = x : X \to L[X]_E$ is the natural inclusion in the category of (Ω, E)-algebras, and
- $\Phi_{L[X]}(\pi_{n+1}(\mu x.p)) = [\psi_n(\mu x.\Phi_{L[X]}(p))/x]\psi_n(\Phi_{L[X]}(p))$ for each $x \in X$ and each $p \in L\{X\}$ (this last using the inductive hypothesis).

Once again this implies that there is an (Ω, E)-algebra map $\psi'_{n+1} : L\{X\}_E \to L[X]_E$ satisfying $\psi'_{n+1} \circ \Phi_{L\{X\}} = \Phi_{L[X]} \circ \pi_{n+1}$, and the conditions just enumerated imply that ψ'_{n+1} satisfies the defining conditions for ψ_{n+1}. Hence the maps again are equal, and so $\psi_{n+1} \circ \Phi_{L\{X\}} = \psi'_{n+1} \circ \Phi_{L\{X\}} = \Phi_{L[X]} \circ \pi_{n+1}$. $\square$

Corollary 7.2. *$\psi_n(L_{cE}) \subseteq L$ for each $n \in \mathbb{N}$.*

Proof. If $p \in L_{cE}$, then there is some $p' \in L_c$ with $\Phi_{L\{X\}}(p') = p$. Then $\psi_n(p) = \psi_n(\Phi_{L\{X\}}(p')) = \Phi_{L[X]}(\pi_n(p')) \in \Phi_{L[X]}(L) \subseteq L$, the membership following from Proposition 2.18. $\square$

Recall that the extended operational model $\mathcal{O}_c : L_c \to O$ is defined by $\mathcal{O}_c(p) = \bigsqcup\{\pi_n(p) \mid n \in \mathbb{N}\}$ (cf. Proposition 5.2), while the extended denotational model $\mathcal{D}_c : L_c \to D$ is defined using the universal algebra structure of L_c (cf. Corollary 4.4).

Corollary 7.3. (i) *Assume the denotational finiteness condition holds. Then, for all $p \in L$ and $q \in L_{cE}$, $\mathcal{D}(p) \leqslant_D \bigsqcup\{\mathcal{D}(\psi_n(q)) | n \in \mathbb{N}\}$ implies that there exists some $m \in \mathbb{N}$ such that $\mathcal{D}(p) \leqslant_D \mathcal{D}(\psi_m(q))$.*

(ii) *Assume the operational finiteness condition holds. Then, for all $p \in L$ and $q \in L_{cE}$, $\mathcal{O}(p) \leqslant_O \bigsqcup\{\mathcal{O}(\psi_n(q)) | n \in \mathbb{N}\}$ implies that there exists some $m \in \mathbb{N}$ such that $\mathcal{O}(p) \leqslant_O \mathcal{O}(\psi_m(q))$.*

Proof. These follow from the corresponding properties for $\mathcal{D}$ and $\mathcal{O}$, and from the preceding result. $\square$

The next result gives both an ordered operational model and an ordered denotational model for L_{cE}. To avoid subscripts on subscripts, we let $\Phi_c : L_c \to L_{cE}$ denote the projection homomorphism on the closed terms.

Proposition 7.4. (i) *There is a $\preceq_{L_{cE}}$-monotone mapping $\mathcal{O}_{cE} : L_{cE} \to O$ such that $\mathcal{O}_{cE} \circ \Phi_c = \mathcal{O}_c$.*

(ii) *There is an $(\Omega\{X\}, E)$-algebra homomorphism $\mathcal{D}_{cE} : L_{cE} \to D$ satisfying $\mathcal{D}_{cE} \circ \Phi_c = \mathcal{D}_c$.*

Proof. (1) Let $p \in L_{cE}$. Then there is some $p' \in L_c$ with $\Phi_{L\{X\}}(p') = p$, and so Corollary 7.2 implies that, for each n, $\psi_n(p) = \Phi_{L[X]}(\pi_n(p'))$. It follows that

$\{\psi_n(p) \mid n \in \mathbb{N}\}$ is an increasing family, and so the same is true of $\{\mathcal{O}(\psi_n(p)) \mid n \in \mathbb{N}\} \subseteq O$. Then, we can define

$$\mathcal{O}_{cE} : L_{cE} \to O \quad \text{by} \quad \mathcal{O}_{cE}(p) = \bigsqcup \{\mathcal{O}(\psi_n(p)) \mid n \in \mathbb{N}\}.$$

To see that $\mathcal{O}_{cE}$ preserves the minimum monotone pre-order, we appeal to Proposition 2.3 and the monotonicity of $\mathcal{O}$.

(2) For $\mathcal{D}_{cE}$, note that since D is an (Ω, E)-algebra, the same is true of D^X and of $[D^X \to D]$, since both of these function algebras are defined pointwise. Then the Ω-algebra homomorphism $\mathcal{D}_{L\{X\}} : L\{X\} \to [D^X \to D]$ factors through the (Ω, E)-algebra $L\{X\}_E$. Thus, there is an (Ω, E)-algebra map $\mathcal{D}_{L\{X\}_E} : L\{X\}_E \to [D^X \to D]$ with $\mathcal{D}_{L\{X\}_E} \circ \Phi = \mathcal{D}_{L\{X\}}$. Then we can define $\mathcal{D}_{cE} : L_{cE} \to [D^X \to D]$ by $\mathcal{D}_{cE} = \mathcal{D}_{L\{X\}_E}|_{L_{cE}}$. Then,

$$\mathcal{D}_{cE} = \mathcal{D}_{L\{X\}_E}|_{L_{cE}} = (\mathcal{D}_{L\{X\}_E} \circ \Phi_c)|_{L_c} = \mathcal{D}_{L\{X\}}|_{L_c} = \mathcal{D}_c,$$

the last equality following from Corollary 4.14. $\square$

Before proving the main theorem of this section, we must consider what it means to substitute a value in a context in the equational setting. We need only consider in detail a fairly special situation. Suppose $C \in L[*]_E$ and $p \in L_{cE}$. Define $C[p] \in L_{cE}$ by

$$C[p] = \overline{\iota_1[* \mapsto p]}C,$$

where $\overline{\iota_1[* \mapsto p]} : L[*]_E \to L_{cE}$ is the unique homomorphism of (Ω, E)-algebras that extends the inclusion $\iota_1 : L \to L_{cE}$ and sends $*$ to p. (The overbar distinguishes it from $\iota_1[* \mapsto p]$, which, according to the notational conventions of Section 2, is an Ω-algebra homomorphism from $L[*]$ to L_{cE}.) If $C' \in L[*]$ and $p' \in L_c$, then we have seen in Section 2 how to define $C'[p'] \in L_c$. It is easy to see that

$$C'[p'] = \iota_2[* \mapsto p']C',$$

where $\iota_2 : L \to L_c$ is the inclusion map.

Proposition 7.5. *Let $C \in L[*]_E$, $p \in L_{cE}$, $C' \in L[*]$ and $p' \in L_c$ be such that $C = \Phi_{L[*]}(C')$ and $p = \Phi_c(p')$. Then*

$$C[p] = \Phi_c(C'[p']).$$

Proof. We use the universal mapping property that defines the Ω-algebra homomorphism

$$\iota_1[* \mapsto p] : L[*] \to L_{cE},$$

followed by an application of Corollary 2.12, to obtain the following chain of equalities:

$$\begin{aligned} C[p] &= \overline{\iota_1[* \mapsto p]}(\Phi_{L[*]}(C')) \\ &= \iota_1[* \mapsto p]C' \\ &= (\Phi_c \circ \iota_2)[* \mapsto \Phi_c(p')]C' \\ &= \Phi_c(\iota_2[* \mapsto p']C') = \Phi_c(C'[p']) \qquad \square \end{aligned}$$

We now prove the main theorem of this paper.

Theorem 7.6. (i) *If $\mathscr{D}$ is order adequate with respect to $\mathscr{O}$ and the denotational finiteness condition holds, then $\mathscr{D}_{cE}$ is order adequate with respect to $\mathscr{O}_{cE}$.*

(ii) *If $\mathscr{D}$ is strongly order fully abstract with respect to $\mathscr{O}$ and the operational finiteness condition holds, then $\mathscr{D}_{cE}$ is strongly order fully abstract with respect to $\mathscr{O}_{cE}$.*

Proof. (1) This follows directly from Theorem 6.1 and the fact that $\mathscr{D}_{cE}$ and $\mathscr{O}_{cE}$ are induced by $\mathscr{D}_c$ and $\mathscr{O}_c$, respectively.

(2) Assume that $\mathscr{D}_{cE}(p) \not\leqslant_D \mathscr{D}_{cE}(q)$ for $p,q \in L_{cE}$. Then there are $p',q' \in L_c$ with $\Phi(p') = p$ and $\Phi(q') = q$, and $\mathscr{D}_{cE}(p) = \mathscr{D}_c(p')$ and $\mathscr{D}_{cE}(q) = \mathscr{D}_c(q')$. As in the proof of Theorem 6.3, we have $\mathscr{D}_c(p') = \bigsqcup_{n\in\mathbb{N}} \mathscr{D}(\pi_n(p'))$, and it follows that there is some $m \in \mathbb{N}$ such that $\mathscr{D}(\pi_m(p')) \not\leqslant_D \mathscr{D}_c(q')$. Then, $\mathscr{D}_c(q') = \bigsqcup_{n\in\mathbb{N}} \mathscr{D}(\pi_n(q'))$ implies that $\mathscr{D}(\pi_m(p')) \not\leqslant_D \mathscr{D}(\pi_m(q'))$. Since $\mathscr{D}$ is strongly order fully abstract with respect to $\mathscr{O}$, there is some context $C \in L[*]$ with $\mathscr{O}(C[\pi_m(p')]) \not\leqslant_O \mathscr{O}(C[q''])$ for any $q'' \in L$ with $\pi_m(q) \preceq_L q''$.

Note that the restriction of $\Phi_{L\{X\}}$ to L is just the inclusion map from L into $L\{X\}_E$. If $q'' \in L_{cE}$ satisfies $q \preceq_{L_{cE}} q''$, then, because $\pi_n(q') \in L$, $\pi_n(q') = \psi_n(q) \preceq_L \psi_n(q'')$ for each $n \in \mathbb{N}$. Hence, $\pi_m(q') \preceq_L \psi_{m+n}(q'')$ for each $n \in \mathbb{N}$, so $\mathscr{O}(C[\pi_m(p')]) \not\leqslant_O \mathscr{O}(C[\psi_{m+n}(q'')])$ for each $n \in \mathbb{N}$. We apply Proposition 2.15 to conclude that $\mathscr{O}(C[\pi_m(p')]) = \mathscr{O}(\pi_m(C[p']))$, and Corollary 2.12 implies that $\mathscr{O}(C[\psi_{m+n}(q'')]) = \mathscr{O}(\psi_{m+n}(\Phi_{L[*]}(C)[q'']))$ for each $n \in \mathbb{N}$. By the operational finiteness condition (part 2 of Corollary 7.3), $\mathscr{O}(\pi_m(C[p'])) \not\leqslant_O \bigsqcup_{n\in\mathbb{N}} \mathscr{O}(C[\psi_{m+n}(q'')]) = \mathscr{O}_{cE}(\Phi_{L[*]}(C)[q''])$. Since $\mathscr{O}(\pi_m(C[p'])) \leqslant_O \mathscr{O}_c(C[p'])$, it follows that $\mathscr{O}_{cE}(\Psi_*(C)[p]) = \mathscr{O}_c(C[p']) \not\leqslant_O \mathscr{O}_{cE}(\Phi_{L[*]}(C)[q''])$. Finally, since $C \in L[*] \subseteq L\{X\}[*]$, C has no variables, and this means that $C \in L_c[*]$, from which it follows that $\Phi_{L[*]}(C) \in L_{cE}[*]$, the (Ω,E)-algebra of contexts over L_{cE}. $\square$

Example 7.7 (*conclusion*). We return to the simple example language we began discussing in Section 2. The base language S whose syntax is given by the BNF-like production rules:

$$p ::= b_S \mid \delta_S \mid \varepsilon_S \mid a \mid p\,;q \mid p+q,$$

so that S is the free (Ω,E) algebra where the signature Ω consists of the constants b,δ and ε, together with the family $\{a \mid a \in A\}$, and has binary operators ; and $+$. The set

E of equational laws that S satisfies also were delineated when S first was introduced. We showed in Section 3, that a cpo D provides both an ordered operational model and an ordered denotational model S. In fact, the two approaches in this case lead to the same model. Another way of saying this is that the operational model is compositional in the sense of being a homomorphism.

We also showed in Sections 4 and 5 that D can be extended to an ordered operational and ordered denotational model for S_c, the closed terms of the language S extended to include identifiers and recursion operators, $\mu x.$ – for each $x \in X$. And, the results of Section 6 imply that this extended denotational model is order adequate and strongly order fully abstract. But, of course, the extended language S_c does not satisfy the equational laws E we gave for S when we first introduced the example.

The results of this section allow us to give semantics for the family of closed terms S_{cE} of the language S extended as an (Ω, E)-algebra. The difference is that in S_{cE}, the operations ; and $+$ are associative, $+$ is commutative and idempotent, ε_S is an identity for ;, δ_S is a left zero for ; and an identity for $+$. And, the same model D again is both an ordered operational model and an ordered denotational model that is order adequate and strongly order fully abstract. This follows since the equational laws we assumed for S hold in D. □

Example 7.8. We now give a brief discussion of the example languages treated in [8]. Recall that Hennessy treats the same simple language $\mathbf{M}_1$ from three distinct operational and denotational points of view. The language has signature Σ, where

- $\Sigma_0 = \{NIL\}$,
- $\Sigma_1 = Act$, where Act is the set of atomic actions,
- $\Sigma_2 = \{+\}$, and
- $\Sigma_n = \emptyset$ for $n > 2$.

The set of equations that the algebra satisfies state that

- $+$ is commutative and associative, and idempotent,
- NIL is an identity for $+$.

Hennessy then introduces three *testing pre-orders*, each of which gives rise to an operational and a related denotational model for the language above. Tests for processes – essentially, running processes in parallel with an experimental process that tests for a successful outcome – are then devised. If the successful outcome is denoted $\top$, and the unsuccessful outcome is denoted $\perp$, and if p is a process and e an experimental test, then we say

$p\ MAY\ e$ if $\top$ is a possible outcome of running p in parallel with e, and

$p\ MUST\ e$ if $\perp$ a *not* a possible outcome of running p in parallel with e.

The three testing pre-orders are defined by

$p \sqsubseteq_{MAY} q$ if and only if, for every test process e, $p\ MAY\ e$ implies that $q\ MAY\ e$.

$p \sqsubseteq_{MUST} q$ if and only if, for every test process e, $p\ MUST\ e$ implies that $q\ MUST\ e$.

$p \sqsubseteq_{EM} q$ if and only if $p \sqsubseteq_{MAY} q$ and $p \sqsubseteq_{MUST} q$.

The MAY pre-order gives rise to the Hoare power domain over $\{\perp, \top\}$, the $MUST$ pre-order gives rise to the Smyth power domain over $\{\perp, \top\}$, and the EM pre-order gives rise to the Plotkin power domain over $\{\perp, \top\}$.

Finite acceptance trees **fAT**, then are defined, and Hennessy shows they can be used to give fully abstract models for the simple language defined above in each of the testing pre-orders. In particular, for processes p and q,

(i) $p \sqsubseteq_{MAY} q$ if and only if $\mathbf{fAT}_W[p] \leqslant \mathbf{fAT}_W[q]$,

(ii) $p \sqsubseteq_{MUST} q$ if and only if $\mathbf{fAT}_S[p] \leqslant \mathbf{fAT}_S[q]$,

(iii) $p \sqsubseteq_{EM} q$ if and only if $\mathbf{fAT}[p] \leqslant \mathbf{fAT}[q]$.

where $\leqslant$ is a partial order on **fAT**, and $\mathbf{fAT}_S$ and $\mathbf{fAT}_W$ are the finite acceptance trees equipped with the *strong* partial order and the *weak* partial order, respectively.

In order to introduce recursion into the language given above, Hennessy is forced essentially to reprove all the results just enumerated for the finitary language for the language of closed terms in the extended language with identifiers and recursion operators added.

The approach we have presented applies to this situation to obtain these results directly. First, we let $\mathbf{M}_1/\equiv_{MAY}$ denote the quotient of $\mathbf{M}_1$ by the equivalence relation the pre-order $\sqsubseteq_{MAY}$ induces, and we endow this quotient with the partial order resulting from the pre-order on $\mathbf{M}_1$. Let $\mathcal{O}_{MAY} : \mathbf{M}_1 \to \mathbf{M}_1/\equiv_{MAY}$ be the quotient map. Since $\mathbf{fAT}_W$ is a denotational model, the natural map $p \mapsto \mathbf{fAT}_W[p]$ is monotone with respect to the minimum monotone pre-order on $\mathbf{M}_1$, and so equivalence (i) above shows that $\mathcal{O}_{MAY}$ also is monotone. In fact, we can conclude that there is an order isomorphism $\mathcal{O}_{MAY}(\mathbf{M}_1) \simeq \mathbf{fAT}_W(\mathbf{M}_1)$, so that, as with our own example language, the operational model is compositional. Now, each of these models can be completed into a cpo (in fact, this is how Hennessy obtains his model for the closed terms), and the image of the language $\mathbf{M}_1$ in each consists of compact elements. It then is routine to show that the finiteness conditions are satisfied. Moreover, order adequacy and strong order full abstraction also are clear. Thus, Theorem 7.6 implies that models completed to cpo's also have this property. Similar arguments apply for $\mathbf{M}_1/\equiv_{MUST}$ and $\mathbf{M}_1/\equiv_{EM}$.

References

[1] K. Apt and G.D. Plotkin, Countable nondeterminism and random assignment, *J. Appl. Comput. Math.* **34** (1986) 724–767.

[2] J.C.M. Baeten and W.P. Weijland, *Process Algebra* (Cambridge Univ. Press, Cambridge, 1990).

[3] J.W. de Bakker, *Mathematical Theory of Program Correctness* (Prentice-Hall International, London, 1980).

[4] J.A. Bergstra and J.W. Klop, An introduction to process algebra, in: J.C.M. Baeten, ed., *Applications of Process Algebra* (Cambridge Univ. Press, Cambridge, 1990) 1–21.

[5] J.A. Bergstra, J.W. Klop and E.-R. Olderog, Readies and failures in the algebra of communicating processes, *SIAM J. Comput.* **17** (1988), 1134–1177.

[6] P.M. Cohn, *Algebra,* Vol. 3 (Wiley, Chichester, 2nd ed., 1991).

[7] H.B. Curry and R. Feys, *Combinatory Logic* (North-Holland, Amsterdam, 1958).

[8] M. Hennessy, *Algebraic Theory of Processes* (MIT Press, Cambridge, MA, 1988).

[9] M. Hyland, A syntactic characterization of the equality in some model for the lambda calculus, *J. London Math. Soc.* **12** (1976) 361–370.

[10] M.W. Mislove, Algebraic posets, algebraic cpo's, and models of concurrency, in: *Proc. Oxford Symp. on Topology* (1991).

[11] M.W. Mislove, L.S. Moss and F.J. Oles, Non-well-founded sets modeled as ideal fixed points, *Inform. Comput.* **93**, 16–54.

[12] M.W. Mislove and F.J. Oles, A topological algebra for angelic nondeterminism, IBM Research Report RC 17344, 1991.

[13] M.W. Mislove and F.J. Oles, A simple language supporting angelic nondeterminism and parallel composition, Lecture Notes in Computer Science, Vol. 598 (Springer, Berlin, 1992) 77–101.

[14] M.W. Mislove and F.J. Oles, Full abstraction and unnested recursion, Lecture Notes in Computer Science, Vol. 666 (Springer, Berlin, 1993) 384–397.

[15] F.J. Oles, *A category-theoretic approach to the semantics of programming languages* Ph.D. Dissertation, Syracuse University, August, 1982.

[16] F.J. Oles, Semantics for concurrency without powerdomains, *Proc. 14th ACM Symp. on the Principles of Programming Languages* (ACM Press, New York, 1987) 211–222.

[17] F.J. Oles, Simultaneous substitution in the lambda calculus, IBM Research Report RC 17596, 1992.

[18] F.J. Oles, When is a category of many-sorted algebras cartesian closed? *Int. J. Found. Comput. Sci.* **3** (1992) 225–231.

[19] V. Stoltenberg-Hansen and J.V. Tucker, Algebraic and fixed point equations over inverse limits of algebras, *Theoret. Comput. Sci.* **87** (1991) 1–24.

[20] E.G. Wagner, Algebras, polynomials, and programs, *Theoret. Comput. Sci.* **70** (1990) 3–34.